

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

NÁVRH ČTYŘ-ROTOROVÉHO VZNÁŠEDLA

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. VÁCLAV UHLÍŘ

BRNO 2012



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

NÁVRH ČTYŘ-ROTOROVÉHO VZNÁŠEDLA

DESIGN OF QUADROCOPTER

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. VÁCLAV UHLÍŘ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JOSEF HÁJEK

BRNO 2012

Abstrakt

Tato studentská práce se zabývá návrhem čtyř-rotorového vznášedla (quadrotor) za účelem akademické platformy pro testování a další vývoj. Součástí práce je základní přehled možností konstrukce, stabilizace a ovládání quadrotorů. Výsledkem práce je koncept quadrotoru a jeho sestrojený prototyp, jehož součástí je i softwarová konzole pro jeho bezdrátové ovládání pod systémem Android.

Abstract

This student paper discusses basic concept of quadcopter for purpose of academic platform for testing and development. Paper includes basic overview of construction, stabilization and controlling of quadcopter. As result of this paper is presented concept of quadcopter and its constructed prototype with included software console for wireless control under Android system.

Klíčová slova

Robotika, UAV, čtyř-rotorové vznášedlo, quadrotor, stabilizace.

Keywords

Robotics, UAV, quadcopter, quadrotor, stabilization.

Citace

Václav Uhlíř: Návrh čtyř-rotorového vznášedla, diplomová práce, Brno, FIT VUT v Brně, 2012

Návrh čtyř-rotorového vznášedla

Prohlášení

Prohlašuji, že jsem tuto semestrální práci vypracoval samostatně pod vedením Ing. Josefa Hájka.

.....

Václav Uhlíř
23. května 2012

Poděkování

Tímto bych chtěl poděkovat Ing. Josefu Hájkovi za odbornou pomoc, věnovaný čas a přístup k potřebné technice.

© Václav Uhlíř, 2012.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	3
2	Čtyř-rotorové vznášedla	4
2.1	Historie	4
2.2	Příklady dnešních quadrotorů	5
2.2.1	Fly & Inspect	5
2.2.2	Quadrotor swarms	5
2.2.3	Datron Scout	6
2.2.4	Projekt Wolfwerl	6
2.2.5	Projekt UAVP-NG	7
3	Návrh konstrukce	8
3.1	Celkový návrh	8
3.2	Mechanické prvky	9
3.2.1	Motory	9
3.2.2	Napájení	10
3.3	Řídící jednotka	13
4	Získávání dat o poloze	14
4.1	Vzdálenostní senzory	14
4.1.1	Senzory využívající Infra-červeného světla	14
4.1.2	Senzory využívající ultrazvuk	15
4.2	Inerciální měřicí jednotka - IMU	16
4.3	Poziční senzory	17
4.3.1	GPS modul	17
4.3.2	Barometrický senzor	18
4.4	Obrazové snímáče	18
4.4.1	Odvození pozice z obrazových dat	19
4.4.2	Hledání cesty a cílů v obrazových datech	19
4.4.3	Příklad kamerového modulu	20
5	Návrh řízení	21
5.1	Stabilizace	21
5.1.1	Stabilizace pomocí PID regulátoru	22
5.1.2	Stabilizace pomocí fuzzy pravidel	23
5.2	Manévrování	24
5.2.1	Let do stran	24
5.2.2	Rotace	25

6	Struktura prototypu platformy	26
6.1	Konstrukce prototypu	26
6.2	Mechanické prvky	27
6.2.1	Motory	27
6.2.2	Vrtule	27
6.3	Hardwarová struktura	28
6.3.1	Řídící jednotka	28
6.3.2	IMU jednotka	29
6.3.3	LCD panel	30
6.3.4	Bluetooth module	31
6.3.5	Regulátory motorů	31
6.4	Softwarová struktura	32
6.4.1	main(.c)	33
6.4.2	btcom(.c/.h)	34
6.4.3	pwm(.c/.h)	35
6.4.4	lcd(.c/.h)	35
6.4.5	chr6d(.c/.h)	36
6.4.6	i2c(.c/.h)	37
6.4.7	adc(.c/.h)	37
7	Komunikace s okolím	38
7.1	Vizuální komunikace	38
7.1.1	Informační diody	38
7.1.2	Display	38
7.2	Akustická komunikace	39
7.3	Datová komunikace	39
7.3.1	Drátová datová komunikace	39
7.3.2	Bezdrátová datová komunikace	39
8	Vzdálené ovládání	43
8.1	Ovládání	43
8.1.1	Sekce pozičních dat	43
8.1.2	Ovládání motorů	44
8.1.3	Testovací parametry	45
8.2	Softwarová struktura	46
9	Konfigurace a testování	47
9.1	Zpracování informací o poloze	47
9.2	Nesouměrnosti modelu	48
9.3	Testování	49
9.3.1	Test celkového tahu	49
9.3.2	Tah motoru potřebný pro změnu náklonu	50
9.3.3	Tahy motorů pro stabilizaci	51
10	Závěr	52
A	Obsah CD	57

Kapitola 1

Úvod

Už od pradávna se lidé snaží si usnadňovat práci všemožnými vynálezy a nástroji. Od původních jednoduchých nástrojů na ulehčení zemědělské práce a lovu zvěře se lidé časem dostali k nástrojům, které jim nejen usnadňovaly práci v mnoha oborech, ale umožňovaly vykonávat činnosti, které by bez těchto nástrojů nebyly možné. Stejně tak se dnes využívají roboti nejen na výpomoc lidem, jako robotické vysavače, sekačky a multifunkční ramena, tak i na provádění úkonů pro člověka velmi problematických nebo dokonce nemožných.

Příkladem takových robotů jsou například letající robotické stroje označované jako UAV (unmanned aerial vehicle - bezpilotní letoun). UAV se dnes využívají převážně v armádní technice, avšak jejich potenciál je vysoký i v mnoha jiných oborech. Příkladem toho můžou být například UAV, které využívají hasiči při monitorování šíření lesních požárů nebo UAV, které zemědělci a geologové používají k výškovému monitorování povrchů. Náklady na vývoj, produkci a provoz však prozatím stále brání hromadnému rozšíření UAV do běžného používání.

Tato práce se zabývá návrhem a konstrukcí UAV v podobě čtyř-rotorového vznášedla určeného jako základní platforma pro další vývoj a testování v této oblasti. Pro umožnění dalšího vývoje na takové platformě bude potřeba při návrhu počítat s možností přidávání dalších modulů a tedy zejména počítat s nutností vyšší nosnosti, možností logického i mechanického připojení dalších modulů a možnosti „snadné“ změny nebo úpravy konstrukce.

Výsledkem práce je návrh konstrukce a rozbor možností využití různých senzorů pro stanovení polohy a náklonu quadrotorů. Dále je výsledkem zhotovený hardwarový i softwarový prototyp quadrotoru dle návrhu a softwarová konzole pro platformu Android určená k bezdrátovému ladění a případně i ovládání.

Jednotlivé části práce jsou rozděleny na historii a současnost quadrotorů - Kapitola 2, základní návrh a možnosti prvků konstrukce - Kapitola 3, přehled jednotlivých typů senzorů využitelných na quadrotoru pro získávání informací o poloze v prostoru a jejich ukázky - Kapitola 4 a rozbor základních možností řízení a stabilizace - Kapitola 5. V Kapitole 6 je následně popsán zhotovený prototyp z hardwarové a softwarové stránky, v Kapitole 7 jsou popsány možnosti práce a komunikace s prototypem dle implementovaných protokolů a Kapitola 8 obsahuje popis implementované konzole pro vzdálené ladění a ovládání. Konfigurace a testování prototypu je popsáno v Kapitole 9 a Kapitola 10 obsahuje zhodnocení dosažených výsledků a jejich porovnání s jinými quadrotory.

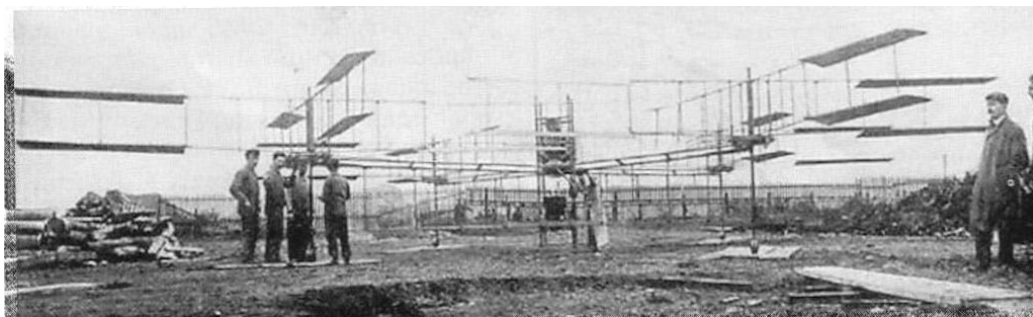
Kapitola 2

Čtyř-rotorové vznášedla

Čtyř-rotorové vznášedlo označované zkráceně také jako quadrotor (spojení slov quad - čtyři a rotor) je zařízení se čtyřmi vertikálními rotory umístěnými nejčastěji v jedné rovině do kříže. Klasická helikoptéra má dva rotory - jeden vertikální zajišťující potřebný vztlak a druhý pomocný horizontální, který kompenzuje točivý moment a umožňuje otáčení kolem svislé osy. Quadrotory využívají většího počtu rotorů, kde dva motory se točí pravotočivě a dva levotočivě, tím se kompenzují rotační síly vytvořené odporem vzduchu na vrtulích a není tedy potřeba horizontálního vyrovnávacího motoru. Čtyři rotory zároveň poskytují dostatečné možnosti pro manévrování.

2.1 Historie

Prvním quadrotorem pravděpodobně byl stroj pojmenován Breguet-Richet Gyroplane No.1 [4]. Tento stroj byl sestaven bratry Louisem a Jacquesem Breguetovými v roce 1907. Stroj sice nebyl schopen samostatného letu, ale udal počátek vývoje helikoptér a podobných strojů s vertikálním startem.



Obrázek 2.1: První pokusy o čtyř-rotorové vznášedlo [6].

Pro běžné pilotované lety se od quadrotorů ustoupilo kvůli potřebné plošné velikosti, kterou zabíraly rotory s vrtulemi. Pro využití jako UAV však tento velikostní problém nevzniká a čtyři rotory nabízejí lepší možnosti stability. Nevýhodou UAV quadrotorů je náročné ovládání, což je hlavní zábranou pro jejich rozšíření do běžného používání.

2.2 Příklady dnešních quadrotorů

Protože dochází k neustálému zmenšování a zlevňování výpočetní techniky, stávají se quadrotory reálným řešením pro řadu problémů. Několik firem se již zabývá komerčním prodejem quadrotorů na dálkové ovládání, které využívají automatické stabilizace pro snadnější ovládání. Tyto quadrotory jsou navíc většinou vybaveny záznamovými video zařízeními nebo dokonce přímo přenášejí obraz bezdrátově. Příklady takových firem mohou být třeba Parrot AR.Drone nebo Draganfly Innovations. Mnoho firem a institucí se ale také zabývá vývojem quadrotorů pro specifické aplikace a pracují na vytváření autonomních quadrotorů, které nebudou ke své činnosti potřebovat zásah člověka.

2.2.1 Fly & Inspect

Quadrotor s názvem „Fly & Inspect“ je příkladem quadrotoru, který je dnes vyvíjen pro reálnou aplikaci. Fly & Inspect je vyvíjen firmou Siemens ve spolupráci s MIT a má sloužit k inspekci budov a prostorů.

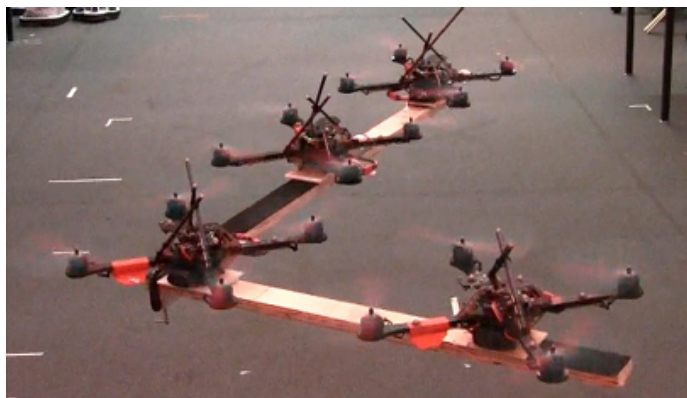


Obrázek 2.2: Quadrotor firmy Siemens [5].

Quadrotor provádí inspekci pomocí laserových dálkoměrů a zároveň zaznamenává okolí kamerami. Zaznamenaná data pak mají sloužit k rozpoznání trhlin, deformací a dalších příznaků opotřebení či poškození skenovaných objektů. Fly & Inspect má provádět let po předem naplánované trase, přičemž se má samostatně vyhnout libovolným překážkám, které mu stojí v cestě, avšak dnes (prosinec 2011) stále quadrotor potřebuje být ovládán člověkem přes dálkové ovládání.

2.2.2 Quadrotor swarms

University of Pennsylvania vyvíjí malé quadrotory, které jsou schopné vzájemné spolupráce například při přenášení objektů nebo při výstavbě konstrukcí [16].



Obrázek 2.3: Quadrotory nesoucí náklad [16].

2.2.3 Datron Scout

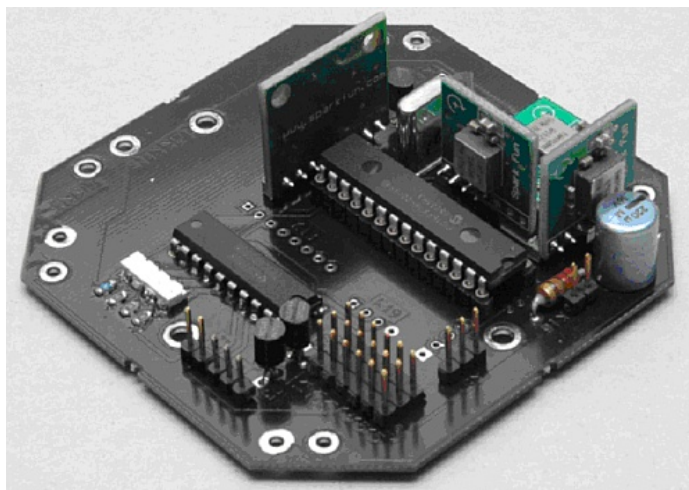
Datron Scout je hotový quadrotor využívaný převážně k vojenským účelům. Jeho hlavní výhodou je skladnost - je možné ho rozložit na malé kompaktní části a jde znovu složit bez potřeby dalšího nářadí. Datron Scout poskytuje instantní videopřenos, navigaci na pozici dle GPS a sledování konkrétních objektů.



Obrázek 2.4: Quadrotor Datron Scout [11].

2.2.4 Projekt Wolfwerl

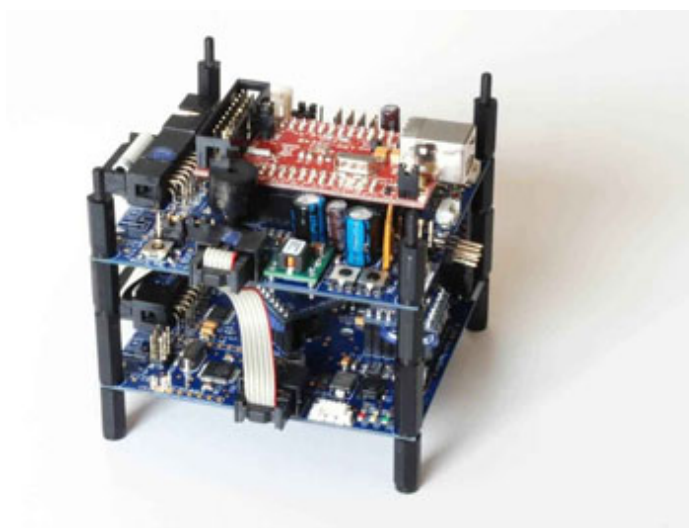
Projek Wolfwerl [9] vznikl v roce 2007 a má za cíl vytvořit opensource platformu quadrotoru dostupné ceny. Na projektu pracují dobrovolníci převážně z Německa, Holandska, Rakouska a Švýcarska. Quadrotor projektu Wolfwerl je určen pro ovládání rádiovým vysílačem a poskytuje základní pomoc v letu stabilizací pomocí PID regulátoru. Výhodou tohoto projektu je relativně snadný elektronický model pro sestavení, který zároveň poskytuje i nějaké základní možnosti zpracování dat ze senzorů. Ovládání motorů quadrotoru je pak možné pomocí PWM modulace nebo přes I²C rozhraní. Kromě ovládání motorů základní deska Wolfwerl umožňuje připojení dvou servo-motorů určených například pro ovládání natočení kamery.



Obrázek 2.5: Základní deska z projektu Wolferl [9].

2.2.5 Projekt UAVP-NG

Projekt UAVP-NG [8] by se dal považovat za nástupce projektu Wolferl. Tento projekt se zaměřuje více na výkon logiky a pro hlavní výpočty využívá procesor ARM7. Na rozdíl od projektu Wolferl je projekt UAVP-NG složen z několika desek, kdy jedna je určená pro řízení letu, další jsou určeny pro senzory a komunikaci. Na projektu pracuje mnoho dobrovolníků, kteří se snaží vytvořit univerzální knihovnu pro různé účely.



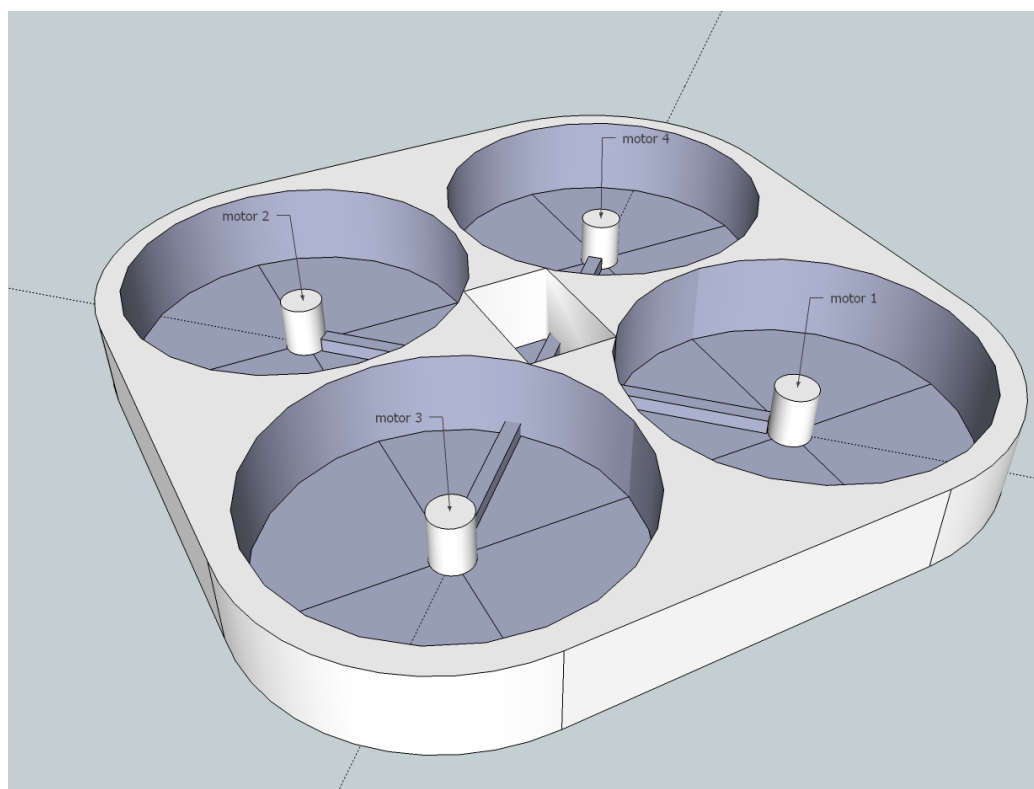
Obrázek 2.6: Desky z projektu UAVP-NG [8].

Kapitola 3

Návrh konstrukce

3.1 Celkový návrh

Základní návrh jsem vytvořil jako jednoduchý quadrotor s křížovým umístěním motorů, kde všechny motory jsou stejně daleko od středu modelu a tedy ideálně i stejně daleko od středu těžiště. Orientace quadrotoru je stanovena tak, že jeden motor je určen jako „přední“ (č. 1), k němu opačný je určen jako „zadní“ (č. 2) a zbylé dva motory můžeme označit za „pravý“ (č. 3) a „levý“ (č. 4) z pohledu ze středu směrem k přednímu motoru.



Obrázek 3.1: Návrh quadrotoru.

Zabírající vrtule vytvářejí odpor vzduchu, který kromě vztlaku způsobuje síly působící do protirotace směru vrtule. Aby nedocházelo k nežádoucím rotacím prototypu, budou dva motory určeny jako pravotočivé a druhé dva jako levotočivé. Kvůli zachování stability při vlivu bočních proudů je vhodné usadit motory se stejnou rotací naproti sobě, tedy na jedné z os budou motory pravotočivé a na ose druhé budou motory levotočivé.

Pro snadnou realizaci a manipulaci bude sestavena platforma quadrotoru střední velikosti, což v tomto případě znamená, že strana půdorysného čtverce bude mít délku zhruba 50 cm. Tato velikost by měla umožnit využití standardní elektroniky, snadnou přenositelnost modelu a tedy optimální platformu pro testování. Zároveň lze předpokládat, že maximální vzletová hmotnost nepřesáhne 20 kg a tedy dle Zákona o civilním letectví [1] tento prototyp nebude považován za letadlo - tedy nebude se muset řídit letovým provozem.

3.2 Mechanické prvky

Mechanické prvky umožňují pohyb zařízení - v tomto případě let. Zatímco například letadla potřebují kromě motorů ještě různé směrovky a klapky, u quadrotorů k získání vztlaku i manévrovatelnosti pro efektivní let a ovládání stačí čtyři motory.

3.2.1 Motory

Pro účely využití na prototypu quadrotoru budeme uvažovat pouze elektromotory, protože jsou snadněji řiditelné, méně údržbové než jiné typy motorů a je jich dostatečné množství v nabídce (jak velikostně, tak výkonnostně) pro uspokojení konkrétních potřeb pro let.

Základní typy elektromotoru můžeme rozdělit na motory napájené stejnosměrným proudem, krokové motory a motory napájené střídavým proudem. Tyto typy motorů se navzájem liší ve způsobu napájení, údržbovosti i využitelnosti.

3.2.1.1 Motory napájené stejnosměrným proudem

Stejnosemnné motory využívají k pohonu stejnosměrný elektrický proud.

Kartáčové stejnosměrné motory (Brushed DC electric motors) jsou jedním z principiálně nejjednodušších typů motorů. Rotační energie je v těchto motorech generovaná pomocí statických magnetů umístěných v loži motorů, mezi kterými rotuje napájená cívka. Napájení na vnitřní rotující cívku se přivádí pomocí kartáčů (odtud název Kartáčové stejnosměrné motory), přičemž rotační cívka s kartáči se chová jako komutátor. Označení kartáčové není příliš přesné, protože dnes se místo drátěných kartáčů většinou používají uhlíkové bloky přitlačené pružinou.

Tyto motory mají výhodu snadného ovládání rychlosti, která je přímo závislá na přivedeném napětí - tedy nastavuje se buď přímo velikostí napětí nebo pomocí PWM. Nevýhodou těchto motorů je, že nejsou bezúdržbové - je potřeba vyměňovat kartáče a samotné motory nemají příliš vysokou životnost.

Bezkartáčové stejnosměrné motory (Brushless DC electric motor) fungují podobně jako kartáčové motory, ale magnety jsou umístěny na rotačním úchytu a cívky jsou statické. Mechanický komutátor je u těchto motorů nahrazen elektronickým řízením a rotační síla je u těchto motorů nepřímo úměrná rychlosti otáčení.

Bezkartáčové stejnosměrné motory mají výhodu delší životnosti, minimální potřebu údržby a vysokého výkonu. Nevýhody těchto motorů jsou hlavně vyšší pořizovací náklady a potřeba kontroléru, který motor ovládá.

3.2.1.2 Krokové motory

Krokové motory jsou obdobou bezkartáčových stejnosměrných motorů, které umožňují přesnou kontrolu nad natočením rotoru.

Krokové motory jsou vhodné pro zařízení, kde je potřebná přesná kontrola otáčení a natočení rotoru. Krokové motory jsou vhodné spíše pro nastavování než hnací výkon. Nevýhody krokového motoru jsou pořizovací cena a nutnost kontroléru.

3.2.1.3 Motory napájené střídavým proudem

Motory napájené střídavým proudem, nazývané také jako asynchronní motory nebo indukční motory, využívají střídavého elektrického proudu, pomocí kterého vytvářejí proměnné indukční pole, kterým otáčejí s rotorem.

Indukční motory mají nevýhodu potřeby střídavého proudu a vytvářejí silné elektromagnetické rušení, díky čemuž jsou nevhodné pro modely.

3.2.2 Napájení

Protože prototyp bude využívat elektromotory, bude potřebovat přísun energie. Nejefektivnější zdroj energie by byl externí zdroj, ale problém při použití externího zdroje je dodávání energie do modelu.

3.2.2.1 Přívodní kabely

Při využití přívodních kabelů bude vznikat problém s narušováním rovnováhy změnou těžiště tažených kabelů, avšak tato metoda může být využitelná pro některé laboratorní testy.

3.2.2.2 Bezdrátový přenos energie

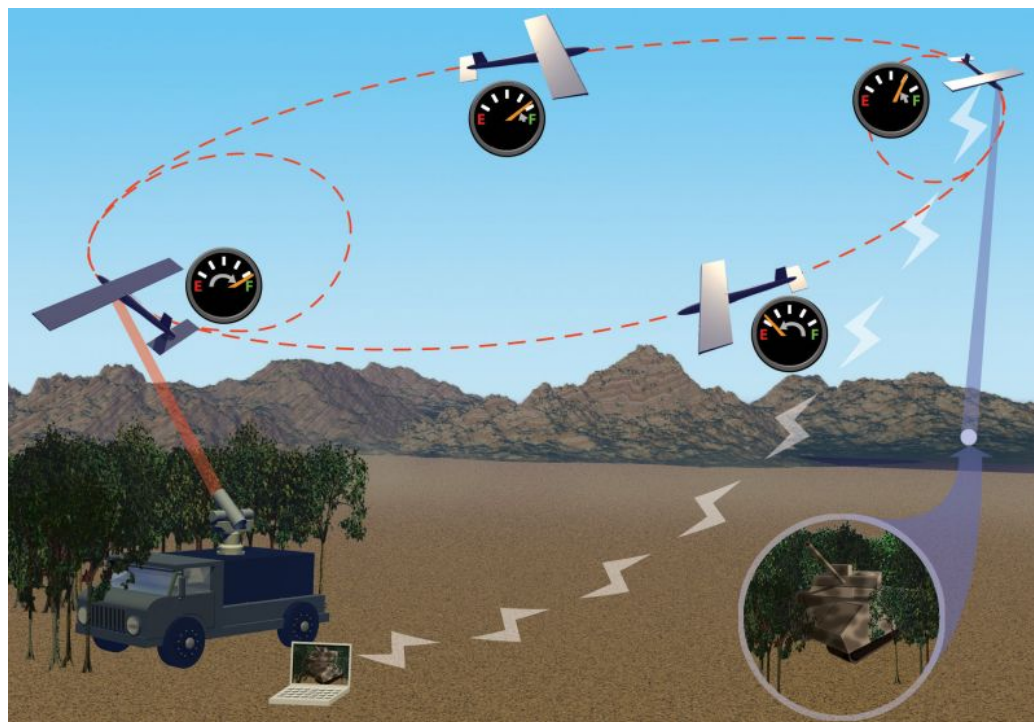
Bezdrátový přenos energie by eliminoval problémy s narušováním rovnováhy taženými kabely, ale přináší větší nároky na energii a rapidně zvyšuje pořizovací i provozní cenu.

Elektrodynamická indukce

Bezdrátový přenos energie je možný například elektrodynamickou indukcí, ale množství přenesené energie významně klesá se vzdáleností. Využití pro model by bylo možné například pro testy ve vyhrazeném prostoru (například místnosti), ale bylo by potřeba využívat opravdu velké vinutí (nejlépe v rámci obvodu místnosti) a požadavky na zdrojovou energii by byly enormní.

Laserový přenos

Další možností bezdrátového přenosu energie je pomocí laserů. Tato technologie je schopná přenášet energii i na větší vzdálenosti, ale potřebuje přímou viditelnost, aby laser mohl cestovat od zdroje k modelu (pole viditelnosti můžeme zvětšit pomocí soustav zrcadel, které umožní viditelnost i v původně stíněných prostorech). Protože je systém založen na přenosu energie pomocí laseru, je také potřeba mít zaměřovací systém, který je schopen laser efektivně směřovat k modelu.



Obrázek 3.2: Dobíjení UAV laserem za letu [14].

Vzhledem k potřebě neustálého přísunu energie motorům quadrotoru by přímé napájení laserem nebylo praktické kvůli nebezpečí výpadků. Laserový přenos energie by však bylo možné efektivně využít jako dobíjení baterií na palubě modelu. Vzhledem k možnostem quadrotorů - držet pozici na určitém místě v prostoru, by bylo teoreticky možné dobíjet energii ze statického nebo pseudo-statického laserového paprsku, čímž by se eliminoval problém nechtěných cílů, které by laser mohl poškodit.

Přenos energie pomocí laserů je zatím stále ve vývoji a je velmi finančně nákladný, ale jeví se jako jedna z pravděpodobných možností budoucího dobíjení UAV strojů za letu. Technologii laserového přenosu energie se například zabývá firma LaserMotive, které se podařilo ve spolupráci s firmou Ascending Technologies sestavit už i funkční prototyp quadrotoru dobíjeného za letu [18].

Využívání přírodních energií

Z přírodních energií je možné využívat například sluneční energii, avšak vzhledem k tomu, že velkou část horizontální plochy na quadrotoru zabírají vrtule, nebylo by možné s aktuálními výkony solárních panelů získávat dostatečné množství energie. Jediné využití by tedy bylo dobíjení baterií ze solární energie a to buď jako prodloužení délky letu (na kvalitě solárních panelů závisí, zda vůbec, případně o kolik, se délka letu prodlouží vzhledem k přidání hmotnosti) nebo jako způsob dobíjení v klidu, tedy že quadrotor přistane, vyčká dobíjení baterií a zase bude moci pokračovat v letu.

3.2.2.3 Baterie

Aby byl quadrotor schopen samostatného letu, bude potřebovat nějaké baterie na palubě. Samozřejmě u baterií panuje problém poměru výkonu a váhy, tedy čím výkonnější baterie na model budeme chtít umístit, tím těžší baterie budou a tím více zatížíme model. Hmotnost baterií samozřejmě závisí také na jejich typu.

Základní rozdělení baterií můžeme udělat na primární a sekundární články. Zatímco primární články fungují na nereverzibilních elektrochemických reakcích, sekundární články, zvané také akumulátory, využívají reverzibilních elektrochemických reakcí a tedy je možné je na rozdíl od primárních dobíjet.

Konkrétní parametry baterií záleží nejen na technologii, ale i na provedení výrobcem a testovacím prostředí, proto se hodnoty v mnoha zdrojích liší. Pokud není uvedeno jinak, orientační hodnoty pro následující srovnání jsou převzaty z [2].

Alkalické články

Alkalické články jsou primární články a tedy jsou nedobíjecí. Jmenovité napětí alkalického článku je 1,5 V a v přepočtu na 1 kg je schopen produkce 110Wh. Alkalické články mají sice nízkou jednotlivou pořizovací cenu, ale vzhledem k nemožnosti dobíjení a tedy nutnosti nakupovat nové, jejich celková cena převyšuje cenu dobíjecí baterie.

NiMH články

NiMH články (Nickel-metal hydride battery) jsou sekundární články se jmenovitým napětím 1,2 V a jsou schopny produkce 95 Wh z 1 kg hmoty. Nevýhodou těchto článků je vysoká úroveň samovybíjení při nepoužívání. Jejich pořizovací cena je vyšší než cena alkalických článků, ale stále patří spíše do kategorie „levné“. Výhodou NiMH článku oproti alkalickým článkům je plošší vybíjecí křivka, tedy jsou schopny při delším vybíjení lépe držet napětí.

Li-ion články

Li-ion (Lithium-ion) články mají jmenovité napětí 3,7 V a jsou schopny produkovat 128 Wh na 1 kg hmoty. Jejich hlavní výhody jsou, kromě vysoké hustoty energie, ve velmi nízkém samovybíjení a vysoké životnosti při opakovaném nabíjení a vybíjení. Nevýhody baterií Li-ion jsou ve „stárnutí“, kdy baterie ztrácí kapacitu nevhledě na používání, a citlivosti na teplo.

Li-S články

Li-S (Lithium–sulfur) články mají jmenovité napětí 1,7 až 2,5 V a jsou schopny produkovat 220 až 300 Wh na 1 kg hmoty [3]. Li-S články bohužel stále trpí relativně rychlou degradací a to zhruba 25% za 30 cyklů [12]. Li-S články by se díky potřebným materiálům a použitým technologiím měly řadit k levnějším bateriím, ale protože stále nejsou sériově vyráběny ve velkém, je jejich pořizovací cena hodně vysoká.

3.3 Řídící jednotka

Quadrotor schopný alespoň částečně autonomního letu bude potřebovat řídící jednotku, která bude zpracovávat data ze senzorů a vypočítávat potřebné nastavení výkonů pro motory. Vzhledem k účelu platformy je potřeba aby řídící jednotka mohla komunikovat se senzory na různých typech sběrnic. Základní podporované sběrnice a protokoly by měly být: I²C, UART, SPI a řídící jednotka by měla být schopná vyčítat data i z analogových senzorů.

Kapitola 4

Získávání dat o poloze

Quadrotor pro základní autonomní řízení potřebuje získat údaje o své poloze a nasměrování. Takovéto údaje se dají například získat z rozpoznávání významných bodů se známou polohou nebo využíváním detekce gravitačních či magnetických polí. Samozřejmě pro lepší řízení je vhodné znát co nejpřesnější informace o aktuální poloze a stavu, čehož se většinou dá dosáhnout kombinací několika metod zjištění údajů o poloze. Kromě samotné polohy je vhodné monitorovat okolí i pro zjištění případných překážek a nebezpečí.

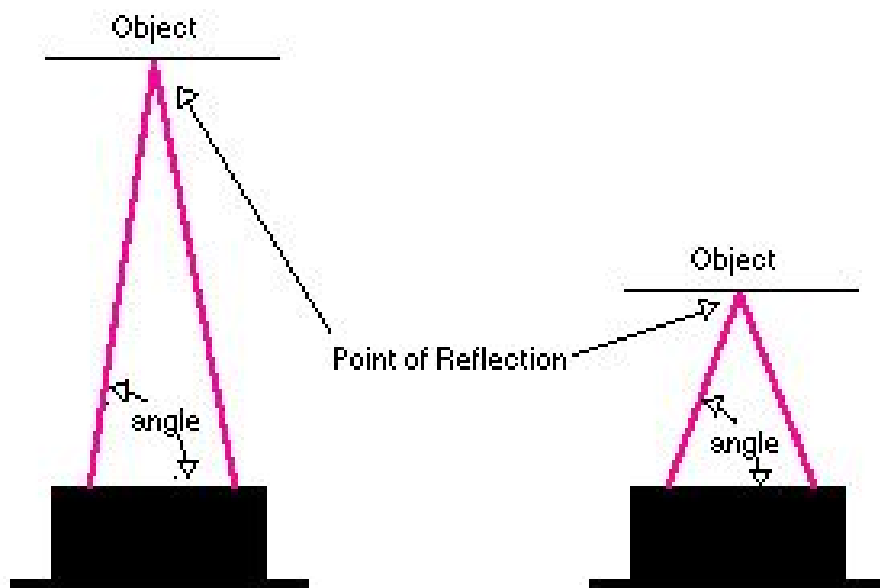
Na platformu byly napojeny pro testování vzdáleností senzory a inerciální jednotka pro zjištění náklonu.

4.1 Vzdálenostní senzory

4.1.1 Senzory využívající Infra-červeného světla

Tato třída senzorů měří vzdálenost pomocí vyslání a příjmu krátkého světelného paprsku, tedy senzor vyše infra-červenou diodou impuls a na malý řádkový CCD (snímač obrazové informace) zachytí odražený paprsek. Podle toho, v které části CCD je paprsek zachycen, senzor dedukuje, pod jakým úhlem paprsek dorazil a tedy může spočítat přibližnou vzdálenost, od jak vzdálené překážky se odrazil.

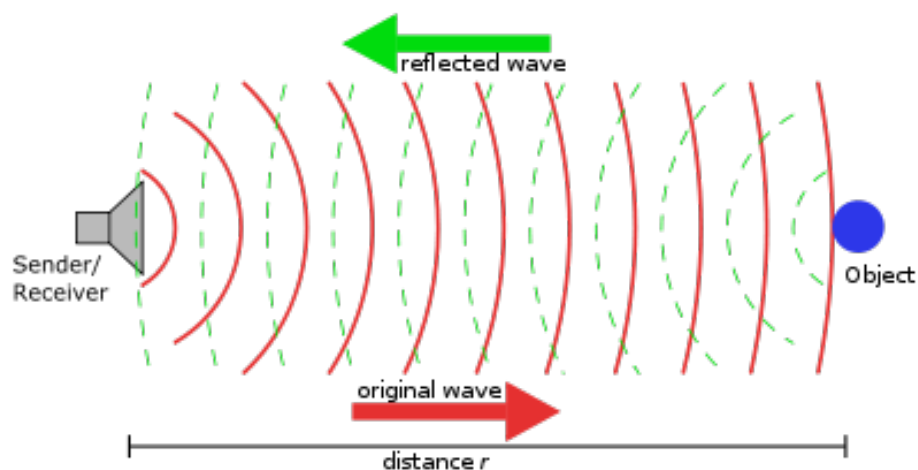
Pro platformu jsme testovali dva typy infračervených senzorů od firmy SHARP: GP2Y0A02 (označován jako senzor pro delší vzdálenosti) s rozsahem 20 až 150 cm [24] a GP2D120 (označován jako senzor pro střední vzdálenosti) s rozsahem 4 až 30 cm [23]. Senzory mají analogový výstup, na kterém naměřenou vzdálenost reprezentují napětím. Výstup ze senzoru je aktualizovaný přibližně každých 40ms. Výstupy jsme přivedli na vstupní porty kanálů 12 bitového AD převodníku přítomného na čipu platformy, kde kontinuálně probíhá převod a výsledky jsou pomocí DMA ihned přístupné pro další zpracování. Původně jsme testovali využití těchto senzorů ke stabilizaci nad vodorovnou plochou, ale výsledky ukázaly, že i při jednolitém povrchu o stejné barvě a odrazuschopnosti naměřená data pro stabilizaci nejsou dostatečně přesná. Využití takovýchto senzorů tedy může být pouze například detekce vzdálenosti prostoru při přistávání, případně detekce vzdálenosti stěn a stropu v místnosti a tedy určení přibližné pozice ve známém prostředí. Pro samotnou detekci překážek tyto senzory vhodné nejsou, protože menší překážky nebo překážky s nízkou odrazuschopností senzor nemusí detekovat.



Obrázek 4.1: Funkce IR senzoru [10].

4.1.2 Senzory využívající ultrazvuk

Senzory využívající ultrazvuk ke zjištění vzdálenosti k překážce, někdy také nazývané Sonary (SOund Navigation And Ranging), využívají znalost rychlosti zvuku a odrazu zvukových vln. Sensor vyšle vlnový signál, který se odrazí od překážky a vrátí se k senzoru se zpožděním rovném času, za jaký zvuk ucestuje dvojnásobek vzdálenosti mezi senzorem a překážkou. Při příjmu odraženého signálu navíc jde pomocí jevu zvaného Dopplerův efekt rozpoznat, zda se detekovaný objekt pohybuje směrem k nebo od senzoru. Pokud se vzdálenost mezi objektem a senzorem zmenšuje, je odražená vlna o vyšší frekvenci než vlna vyslaná a naopak.

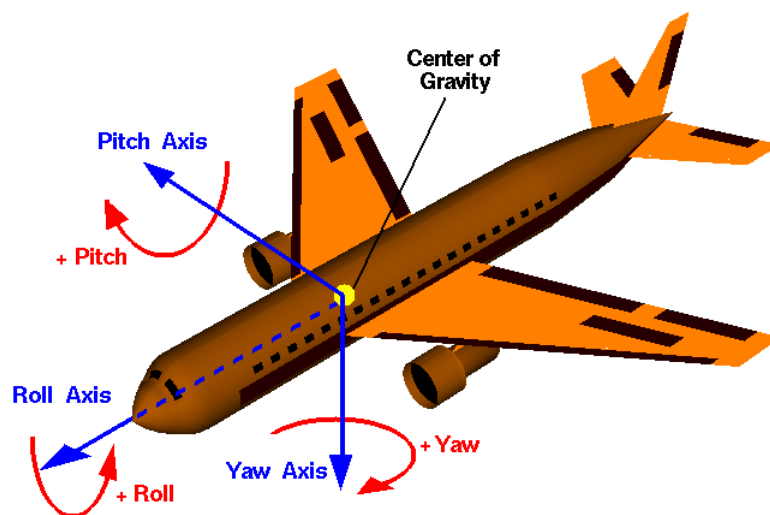


Obrázek 4.2: Funkce ultrazvukového senzoru [31].

Pro platformu jsme testovali senzor SRF10. Jeho maximální dosah je 6m a komunikuje po sběrnici I^2C . Při detekci objektů na větší vzdálenosti klesá obnovovací frekvence výsledků na 65 ms a více. Sérií testů jsme tedy určili nastavení maximálního dosahu na 200 cm (což zajišťuje dostatečnou obnovovací frekvenci pro filtrování a včasnou detekci - zhruba 8 ms) a tedy maximální analogové zesílení jsme nastavili na hodnotu 140 (z rozsahu 40 až 700), aby senzor nezachytával ozvěnu předchozího měření od vzdálenějších překážek. Tento typ senzorů detekuje objekty nacházející se v kuželu před senzorem, přičemž vrací vzdálenost nejbližšího detekovaného objektu a je tedy vhodný hlavně na detekci překážek v dráze letu.

4.2 Inerciální měřicí jednotka - IMU

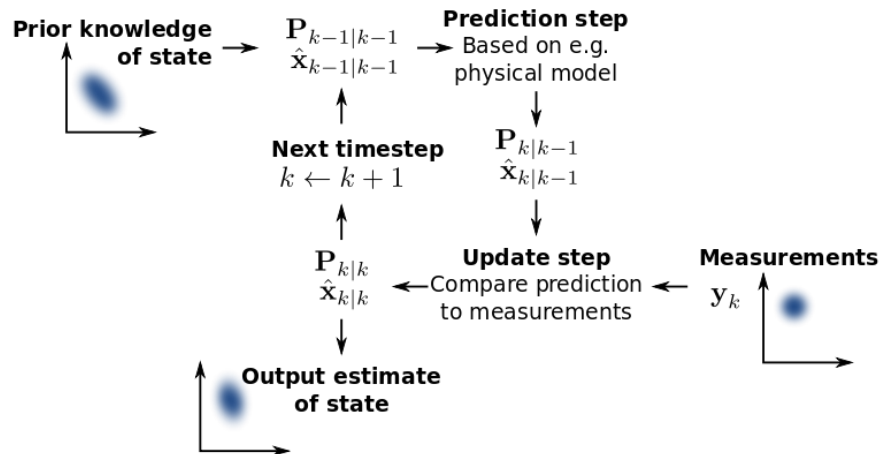
Inerciální měřicí jednotka je určena pro potřeby zjištění náklonu a to jak aktuálního, tak relativního v čase - tedy i pohybu. IMU se standardně skládá z akcelerometrů a gyroskopů. Pro vývoj platformy quadrotoru jsme využili CHR-6D Inertial Measurement Unit, která obsahuje tříosý akcelerometr s rozsahem $\pm 3g$ pro určení zrychlení ve třech osách (někdy označované jako G-forces) a tříosý gyroskop s rozsahem $\pm 400^\circ/s$ pro určení rotace od určitého stavu. Výstupem IMU CHR-6D jsou nejen aktuální data z gyroskopů a akcelerometrů ale i přímo vertikální rotace v příčné ose modelu (Pitch) a vertikální rotace v ose modelu (Roll).



Obrázek 4.3: Znázornění rotací Yaw, Pitch a Roll [17].

CHR-6D je k platformě připojen pomocí protokolu UART a je možné ho paralelně připojit i k počítači na sledování dat pro ladění a optimalizace.

Výpočet úhlů vertikálních rotací se zpravidla u Inerciálních měřicích jednotek využívá Kalmanův filtr [13]. Pomocí tohoto filtru je možné výrazně zpřesňovat zašuměná data snímávaná v čase. Kalmanův filtr využívá modelu, kde si uchovává interní stav filtrované proměnné a pravděpodobnost, že je tento údaj přesný. V jednotlivých krocích pak filtr odhaduje následující stav a ten porovnává s nově naměřenými hodnotami. Výstupem každého kroku je pak nový stav a pravděpodobnost jeho přesnosti.



Obrázek 4.4: Schéma modelu měření za pomoci Kalmanova filtru [30].

Při hledání dat o náklonu z akcelerometrů a gyroskopů Kalmanovým filtrem se využívá dat z akcelerometrů pro zpřesňování aktuálního náklonu a krok systému se provádí podle dat z gyroskopů. Nastavením „důvěryhodnosti“ senzorů poté lze například určit, že se má více věřit akcelerometrům - což je vhodné hlavně pro určení polohy v klidu nebo při plynulých posunech. Oproti tomu při měření vibrujících či prudce se pohybujících systémů je vhodné klást důraz spíše na data z gyroskopů, protože data z akcelerometrů budou pravděpodobně zašumělá a nepřesná.

4.3 Poziční senzory

Informace o pozici je možné v rámci místností získávat ve speciálních případech pomocí vzdálenostních senzorů, avšak obecně vzdálenostní senzory neumožňují zjištění absolutní pozice.

4.3.1 GPS modul

GPS neboli „Global Positioning System“ je systém, který umožňuje určení pozice na základě informačních signálů, které získává ze satelitů obíhajících Zemi ve výšce středního orbitu (zhruba 20200 km nad povrchem [27]). GPS zařízení pak určují pozici pomocí trilaterace, kdy zaznamenají čas příchodu zpráv od jednotlivých satelitů a pomocí zpoždění odvodí pozici. Pro určení pozice na povrchu země stačí tři satelity a pro pozice ve vzduchu pak satelity čtyři. GPS zařízení jsou pak schopna určit v otevřeném prostranství pozici zhruba s přesností okolo 4 m. Vyšší přesnosti lze dosáhnout například využitím systému DGPS, kdy se používá ke korekci odchylka naměřená referenčním přijímačem o stanovené pozici.

Pro využití na platformě jsem testoval GPS Pod od firmy SUUNTO jehož součástí je GPS modul Fastrax IT03-S. Výhodou tohoto modulu je možnost komunikace pomocí protokolu MNEA nebo iTalk na sběrnici UART.

Na desku GPS Podu jsem napájel konektor pro vyvedení RT, TX a GND portů UARTu a ze zařízení je nejen možné sledovat aktuální data, ale lze i data nechat zaznamenávat a poté vyčíst celý záznam o uskutečněné trase.



Obrázek 4.5: Rozmontovaný SUUNTO GPS Pod.

4.3.2 Barometrický senzor

Barometrický senzor slouží primárně k měření atmosférického tlaku. Jeho využití je v robotice především na zpřesnění výškové polohy získané od GPS přijímače, stabilizace v úrovni se stejným tlakem či odhad výšky od místa startu.

Problém určování výšky barometrickým měřením spočívá ve faktu, že atmosférický tlak se nemění pouze s výškou, ale i s teplotou a vlhkostí vzduchu. V krátkodobém měřítku se však dá předpokládat, že vlastnosti atmosférického tlaku závislé na počasí budou nevýznamné a rychlost stoupání či klesání v řádech metrů se dá odvodit z rovnice změny tlaku:

$$\Delta P = \rho * g * (\Delta h), \quad (4.1)$$

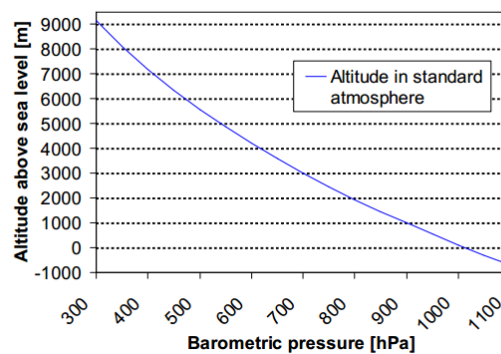
kde ρ reprezentuje hustotu a g gravitaci. Tedy například při teplotě 20°C a nadmořské výšce 300 m je hustota vzduchu zhruba $1,159 \text{ kg/m}^3$ a při dosazení gravitační konstanty $g = 9,81 \text{ m/s}^2$ dostaneme:

$$\Delta h = \frac{\Delta P}{1,159 * 9,81}. \quad (4.2)$$

Příkladem barometrického senzoru pro využití na quadrotoru může být senzor BMP085 firmy Bosch [7]. Výhodou tohoto senzoru je malá velikost, nízká spotřeba a cenová dostupnost. Senzor komunikuje na rozhraní I²C a je schopen odezvy za 4,5 až 25,5 ms podle nastavení požadavku přesnosti, která se v přepočtu pohybuje zhruba od 0,5 do 0,1 m. Senzor nabízí i odhad celkové nadmořské výšky podle interního vzorce.

4.4 Obrazové snímáče

K získávání informací o poloze mohou sloužit i obrazové snímáče. Konkrétně je například možné rozpoznávat v obraze určité předem definované objekty nebo tvary a určit rela-



Obrázek 4.6: Graf znázornění odhadu výšky interním výpočtem senzoru BMP085 [7].

tivní pozici od těchto objektů. Další možností využívání obrazových dat je rozpoznávání překážek, hledání cesty či cílů. Tyto metody jsou užitečné především pro vytváření autonomních systémů.

4.4.1 Odvození pozice z obrazových dat

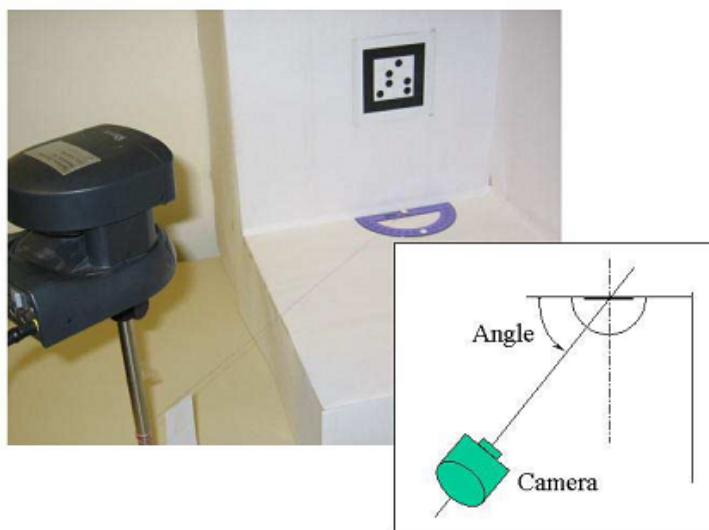
Odvození pozice z obrazových dat je možné při rozpoznání předem definovaných objektů se známou pozicí. Pro tento účel je vhodné například využívat dvou-dimenzionálních (2D) značek či přímo QR kódů. Využívání 2D značek má v běžném prostředí výhodu snadné detekce a zároveň možnost získání informace o značce, ke které systém zná pozici.

Samotné rozpoznání relativní pozice kamery k nalezené značce pak závisí na kvalitě obrazu a vzdálenosti značky. K vypočtení pozice se pak používají stejné metody jako pro vy počítávání transformačních matic na zobrazování objektů v aplikacích s využitím virtuální reality. Pro obecné řešení problému detekce značek a jejich pozice existuje řada knihoven, jejichž součástí je řešení tohoto problému, jejich přehled a srovnání je možné například najít v [32].

4.4.2 Hledání cesty a cílů v obrazových datech

Obrazová data jsou pro robotiku cenným zdrojem informací o okolí ve smyslu využití k detekci překážek, hledání cesty či hledání cílů. Problém quadrotorů spočívá v možnosti pohybu do všech směrů a tak na rozdíl například od vozidel nestačí mít jednu statickou kameru směrem „dopředu“. Systém řízení quadrotoru je samozřejmě možné nastavit tak, že se bude pohybovat kupředu vždy stejnou stranou, avšak je to zbytečné omezení a zpomalení letu a zůstává problém se sledováním prostoru nad a pod quadrotorem.

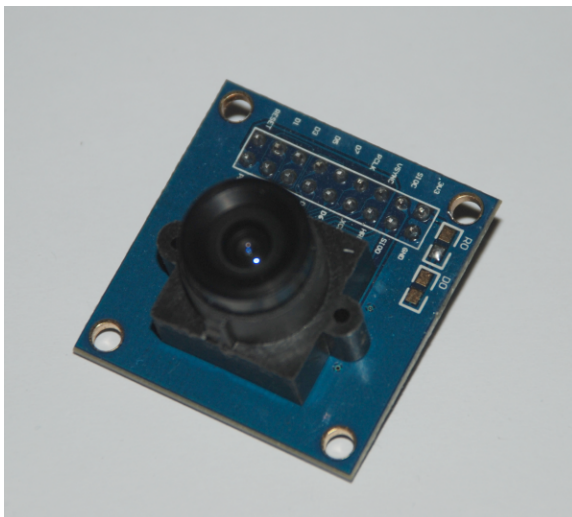
Detekce překážek kamerovým systémem tedy pro létající modely je problematická a vyžaduje buď vícero kamer a tedy vysoký výkon na zpracování dat, či zpomalení řízení manévry navíc. Lepších výsledků je možné dosahovat v oblasti hledání cílů, kde příkladem využití může být hledání vyznačených míst pro přistání, kdy stačí jedna kamera nasměrovaná pod quadrotor.



Obrázek 4.7: Znázornění rozpoznání relativní pozice ke značce [32].

4.4.3 Příklad kamerového modulu

Příkladem kamerového modulu pro základní využití na quadrotoru může být modul s čipem VGA OV7670 [19]. Tento modul se vyznačuje nízkou spotřebou, nízkou hmotností, malou velikostí a dostupnou cenou. Nevýhodou tohoto modulu je rozlišení „pouze“ 640x480 pixelů a rychlostí 30 snímků za vteřinu a tedy bude efektivně použitelný spíše na kratší vzdálenosti. OV7670 disponuje komunikačním protokolem I²C a je možné z něj obraz vyčítat v řadě různých standardů.



Obrázek 4.8: Kamerový modul s čipem OV7670.

Kapitola 5

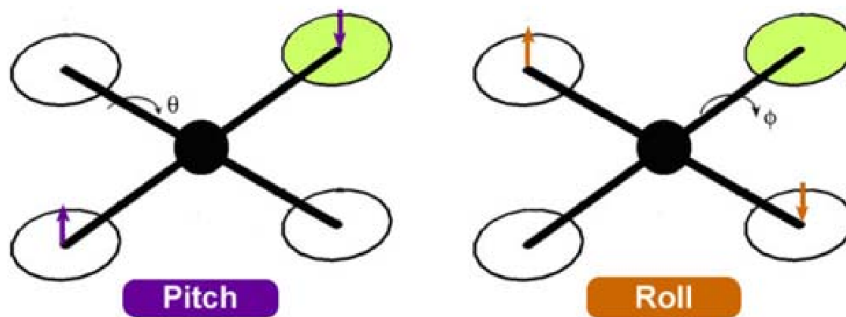
Návrh řízení

Pro řízení quadrotoru je v první řadě nutná stabilizace, tedy zjištění aktuální polohy a stavu quadrotoru a provedení akcí vedoucích k změně stavu a polohy na definované hodnoty. Po úspěšné stabilizaci je pak možné quadrotor snadno řídit. Pokud je řízení správně implementováno, výhoda čtyř rotoru spočívá ve vysoké manévrovatelnosti.

5.1 Stabilizace

Vzhledem k dostupným prvkům jsem se rozhodl pro stabilizaci quadrotoru s využitím dat z IMU CHR-6D. Jednotka CHR-6D s novějším firmwarem (v.2) poskytuje nejenom data z tříosého akcelerometru a data z tříosého gyroskopu, ale provádí i zpracování vstupních dat pomocí rozšířeného Kalmanova filtru konkrétně Hybridního Kalmanova filtru určeného pro fyzikální modely ve spojitém čase s využitím diskrétně měřených hodnot a na výstup dodává přímo rotace ve třech osách od kalibrované polohy. Implementace Hybridního Kalmanova filtru je provedena dle [28].

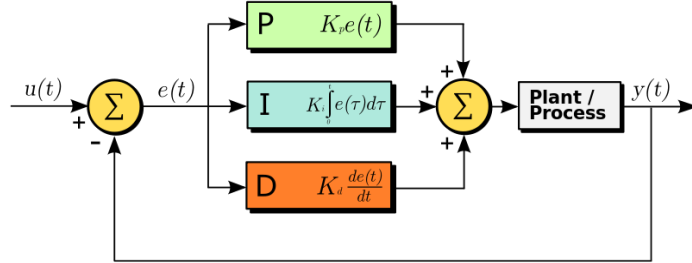
Samotná data o náklonu je však ještě potřeba nějak zpracovat a aplikovat na motory. Pro začátek se budeme zabývat vychýlením Roll a Pitch - tedy vychýlením ve vodorovných osách. Předpokládejme, že quadrotor je vyvážený a že motory společným výkonem dodávají vztlak potřebný pro určitou stabilní výšku. Pokud nyní dojde k vychýlení v některé z vodorovných os, je potřeba zvýšit výkon motoru, který byl vychýlen pod původní osu a pro zachování celkového točivého momentu a vztlaku snížit výkon motoru k němu opačného.



Obrázek 5.1: Znázornění provedení náklonu Pitch a Roll [21].

5.1.1 Stabilizace pomocí PID regulátoru

PID regulátor [29] slouží k řízení systémů se spojitými veličinami s využitím zpětné vazby. Vstupem takového regulátoru je nejenom veličina, podle které chceme systém řídit, ale i aktuální naměřený výstup, dle kterého určujeme aktuální chybu. Samotné části PID regulátoru se skládají z Proporcionální, Integrační a Derivační složky, kde:



Obrázek 5.2: Schéma PID regulátoru [29].

- Proporcionální složka určuje přímou míru vlivu chyby na výstup za pomoci konstanty zesílení označované jako K_p . Výstup proporcionální složky tedy je:

$$P_{\text{out}} = K_p e(t), \quad (5.1)$$

kde $e(t)$ vyjadřuje vstupní hodnotu chyby.

- Integrační složka určuje míru vlivu chyby na výstup podle velikosti i trvání chyby, tedy pomocí integrálu chyby a výsledná hodnota je násobena integračním koeficientem K_i , tedy:

$$I_{\text{out}} = K_i \int_0^t e(\tau) d\tau, \quad (5.2)$$

- Derivační složka určuje míru vlivu chyby na výstup pomocí derivace chyby a tedy určením změny chyby v čase. Tato složka slouží ke zpomalení rychlosti změny aplikované na systém, aby nedocházelo k oscilaci okolo požadované hodnoty a vylepšuje celkovou stabilitu. Derivační složka je zase násobena derivačním koeficientem K_d , tedy:

$$D_{\text{out}} = K_d \frac{d}{dt} e(t) \quad (5.3)$$

Celkový výstup $y(t)$ regulátoru tedy dostaneme jako:

$$y(t) = P_{\text{out}} + I_{\text{out}} + D_{\text{out}}, \quad (5.4)$$

tedy po dosazení z rovnic 5.1, 5.2 a 5.3 dostaneme:

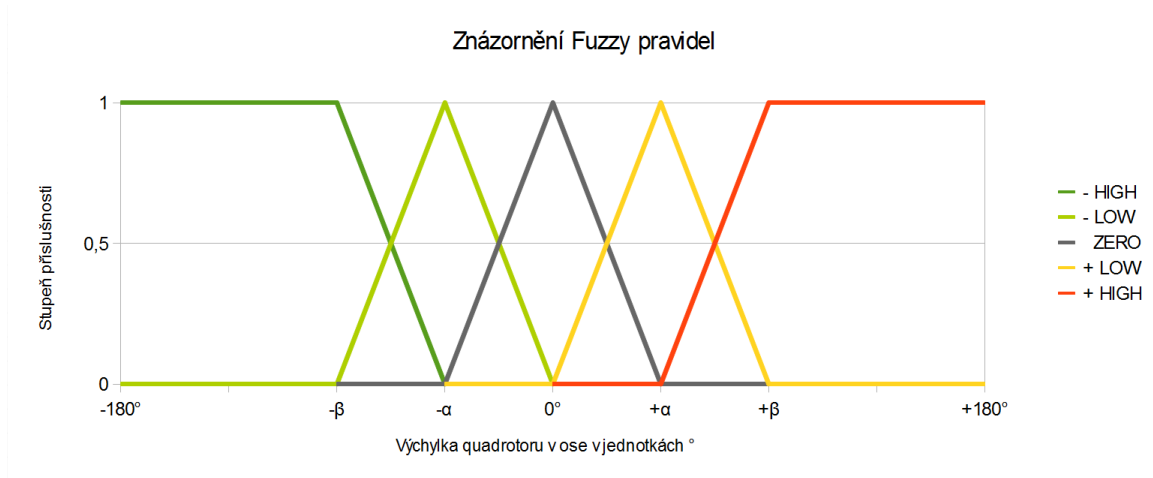
$$y(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t) \quad (5.5)$$

5.1.2 Stabilizace pomocí fuzzy pravidel

Pro stanovení vztahu náklonu v jednotlivých vodorovných osách a výkonu motorů v osách k nim odpovídajících jsem se rozhodl otestovat Fuzzy řízení. Konkrétně zavést 5 pravidel pro každou z vodorovných os. Pravidla jsou označena jako „-HIGH“, „-LOW“, „ZERO“, „+LOW“ a „+HIGH“, kde pro jednotlivá pravidla platí:

- pravidlo „-HIGH“ se plně aplikuje při úhlu -180° až $-\beta$ a proporcionálně do úhlu $-\alpha$
- pravidlo „-LOW“ se aplikuje proporcionálně od úhlu $-\beta$, plně při úhlu $-\alpha$ a proporcionálně do úhlu 0°
- pravidlo „ZERO“ se aplikuje proporcionálně od úhlu $-\alpha$, plně při úhlu 0° a proporcionálně do úhlu $+\alpha$
- pravidlo „+LOW“ se aplikuje proporcionálně od úhlu 0° , plně při úhlu $+\alpha$ a proporcionálně do úhlu $+\beta$
- pravidlo „+HIGH“ se aplikuje proporcionálně od úhlu $+\alpha$ a plně při úhlu $+\beta$ až 180°

kde α a β jsou předem definované úhly, pro které platí $0^\circ < \alpha < \beta$. Každé pravidlo má výstup Δp , který určuje změnu, která se má aplikovat na motory v dané ose, přičemž lze očekávat, že pro pravidlo „ZERO“ platí $\Delta p_{zero} = 0$ a pro pravidla „-HIGH“ a „+HIGH“ respektive „-LOW“ a „+LOW“ platí $\Delta p_{-high} = -\Delta p_{+high}$ respektive $\Delta p_{-low} = -\Delta p_{+low}$.



Obrázek 5.3: Znázornění fuzzy pravidel pro jednu osu.

Příklad: Předpokládejme, že se quadrotor vychýlí v příčné vodorovné ose Pitch o úhel α . V tomto případě se použije pouze jediné pravidlo a to pravidlo „+LOW“ a motor 1, respektive motor 2, se tedy nastaví na $p_1 = p_{base} - \Delta p_{+low}$ respektive $p_2 = p_{base} + \Delta p_{+low}$.

Protože při použití jen aktuálního náklonu by při zpoždění docházelo k překmitům přes vodorovnou polohu, je implementována sada dalších pěti pravidel pro každou z vodorovných os. Vstupem těchto pravidel je změna náklonu v čase a aplikuje sílu do protipohybu quadrotoru. Tedy pokud je quadrotor vychýlen do $+Pitch$ první sada pravidel má odezvu tahu směrem do $-Pitch$, ale ve chvíli, kdy se quadrotor bude blížit do stabilní polohy moc

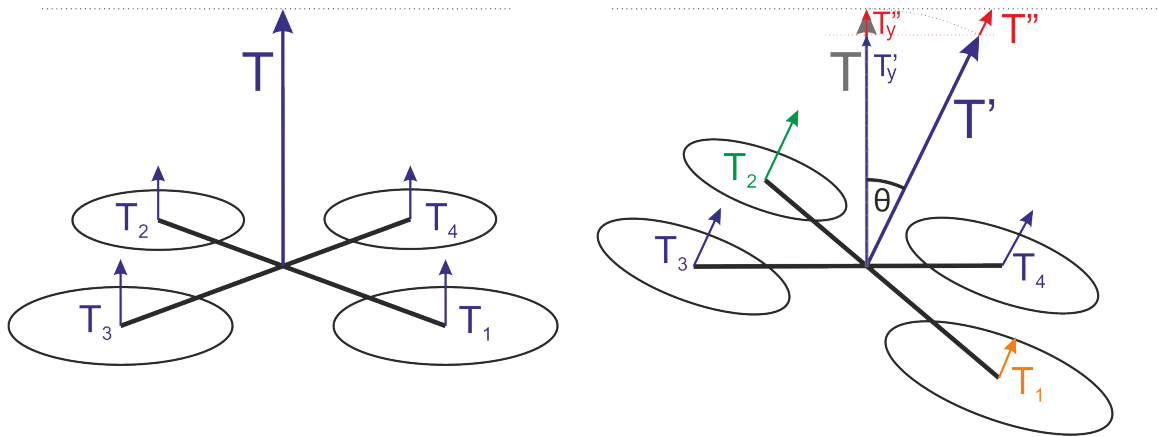
rychle, začne zabírat druhá sada pravidel směrem do $+Pitch$, aby se quadrotor nepřeklopil do $-Pitch$.

5.2 Manévrování

Pokud se quadrotor umí stabilizovat, můžeme přistoupit k základnímu manévrování - tedy let do stran (let vpřed, let vzad a let do stran) a rotacím ve vertikální ose - Yaw.

5.2.1 Let do stran

Let do stran se u quadrotoru zařídí nakloněním ke straně, na kterou chceme letět. Pro naklonění aplikujeme podobný mechanismus jako u stabilizace, tedy snížíme výkon motoru na straně, kam chceme letět a zvýšíme výkon motoru na straně opačné. Pokud zároveň chceme zachovat výšku quadrotoru, je potřeba ještě navíc patřičně zvýšit výkon všech motorů, protože část vztlačové síly produkované motory byla přesunuta k pohybu do boku.



Obrázek 5.4: Znázornění potřeby korekce pro udržení výšky.

Původní vztlačová síla T se nakloněním quadrotoru o úhel θ změní na sílu T' , jejíž vertikální složka T'_y je menší než T . Pokud tedy chceme udržet výšku, musíme zvýšit celkový výkon motorů o T'' , kde víme, že platí:

$$T = T'_y + T'', \quad (5.6)$$

tedy:

$$T'' = T - T'_y, \quad (5.7)$$

a po natočení quadrotoru pro nový celkový tah tedy platí:

$$T'' = T * \sec(\theta) - T'_y * \sec(\theta), \quad (5.8)$$

přičemž víme, že:

$$T'_y = T * \cos(\theta), \quad (5.9)$$

a tedy:

$$T'' = T * \sec(\theta) - T * \cos(\theta) * \sec(\theta), \quad (5.10)$$

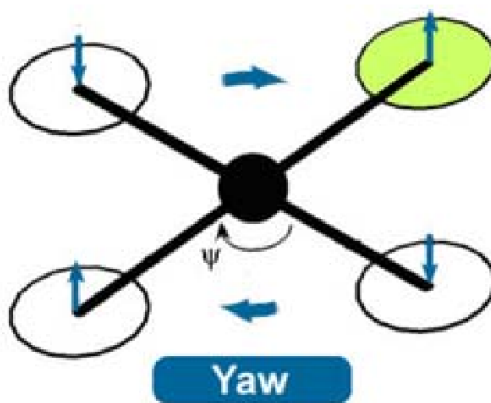
kde po zjednodušení dostaneme potřebné zvýšení tahu jako:

$$T'' = T * (\sec(\theta) - 1). \quad (5.11)$$

Potřebné zvýšení tahu T'' je pak potřeba rozložit pro jednotlivé motory a to jako navýšení tahu každého motoru o $\frac{1}{4}T''$.

5.2.2 Rotace

Rotace dle vertikální osy - Yaw se dosahuje zvýšením výkonu motorů na jedné z os. Protože motory umístěné na stejné ose mají stejný směr rotace, zvýšením potažmo snížením jejich výkonu dochází k rozdílu točivých momentů jednotlivých os a tedy celkový součet točivých momentů je nenulový a díky odporu vzduchu na vrtulích dochází k rozdílným rotačním odporovým silám a dochází k rotaci celého modelu.



Obrázek 5.5: Znázornění dosažení rotace Yaw [21]

Pro zachování výšky (tedy celkového vztlaku) bude rotace prováděna souměrným zvýšením výkonu na jedné ose a snížením na ose druhé.

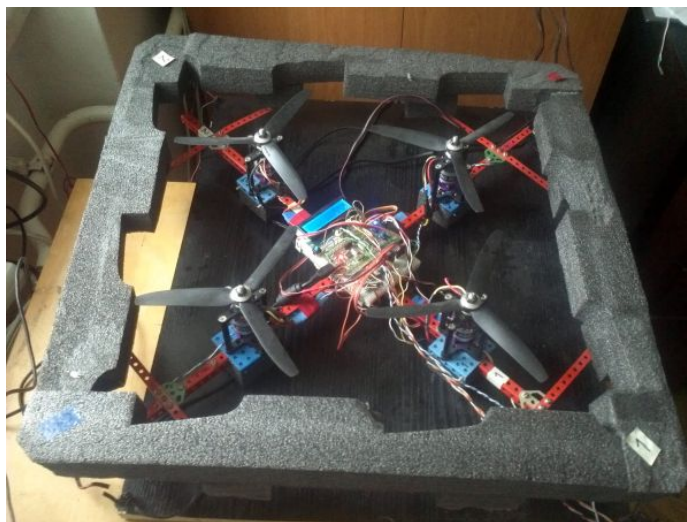
Kapitola 6

Struktura prototypu platformy

6.1 Konstrukce prototypu

Vzhledem k charakteru platformy jsem se rozhodl konstruovat prototyp z prvků umožňujících modularitu a možnost snadných úprav. Tyto požadavky splňuje stavebnice Merkur. Jednotlivé části se skládají z kovových plátů, které jsou děrované ve vzdálenosti 1 cm a tedy poskytují množství kombinací úchytů a zároveň jsou relativně lehké. Bohužel, četné děrování má za důsledek také menší pevnost a jednotlivé části se při kolmém tlaku ohýbají. Tento problém však na místech požadujících vyšší pevnost (např. vzpěry motorů) lze řešit využitím vzpěr s profilem tvaru „L“, případně s profilem ve tvaru „U“, které vykazují mnohem menší ohebnost. Jednotlivé části jsou pak spojovány klasickými šroubky, kde kvůli vibracím bude vhodnější na potřebných místech místo standardních matic použít samojistící matice s polyamidovou vložkou, aby nedocházelo k uvolňování spojů.

Dle návrhu jsem zkonstruoval kostru křížového tvaru, kde rozpětí protilehlých motorů činí 310 mm. Pro zabezpečení okolí i prvků prototypu jsem prototyp vybavil Polyethylenovým hrazením, které pokrývá celý obvod prototypu a slouží i jako tlumení nárazů při nouzových přistáních.



Obrázek 6.1: Osazený prototyp quadrotoru.

Celkové zevní rozměry prototypu jsou čtvercový půdorys 565 mm, tedy úhlopříčka 780 mm a výška 155 mm. Hmotnost prototypu i s veškerou elektronikou se pohybuje okolo 1647 gramů.

6.2 Mechanické prvky

6.2.1 Motory

Na prototypu platformy jsou použity motory MIG RC 400/10 TURBO značky Megamotor. Jedná se bezkartáčové elektromotory na napětí 12 V a spotřebou až 20 A. Motory jsou určeny jak pro větroně a letadla, tak i pro 3D modely. Doporučená zátěž na jeden motor pak činí 550 až 750 g. Výhodou těchto motorů je vysoký výkon (1150 otáček za minutu na jeden volt), nevýhodou je vyšší hmotnost (76 g) a spotřeba, což však pro testovací platformu není tak podstatné.



Obrázek 6.2: Motor MIG RC 400/10 [25].

6.2.2 Vrtule

Jako vrtule pro prototyp byly zvoleny vrtule HD-8040x3 firmy GWS. Jedná se o třílisté vrtule o poloměru 103 mm. Třílisté vrtule mají oproti dvoulistým vrtulím výhodu většího záběru a pro dosažení potřebného vztlaku tedy stačí vrtule menší.

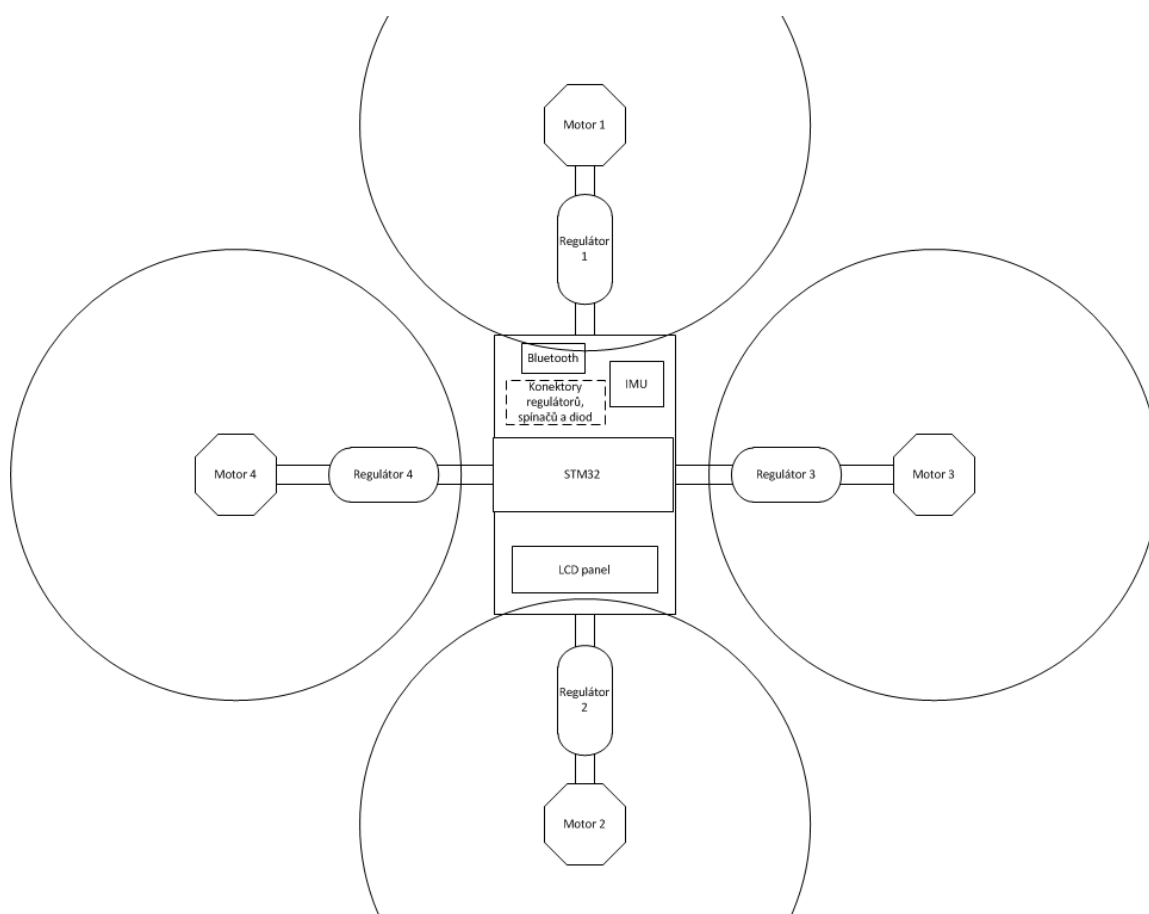


Obrázek 6.3: Vrtule GWS HD-8040X3.

6.3 Hardwarová struktura

Hardwarová struktura byla navržena jako kompromis mezi co nejmenší celkovou plochou a dostatkem místa pro základní elektroniku. Většina elektroniky je umístěná na střední části platformy a regulátory jsou umístěny na vzpěrách k motoru.

Jelikož je střední část již zaplněná, nastává problém s přidáváním další elektroniky. V případě potřeby přidání dalších prvků by tedy bylo potřeba restrukturalizovat stávající elektroniku či vytvořit další patro.

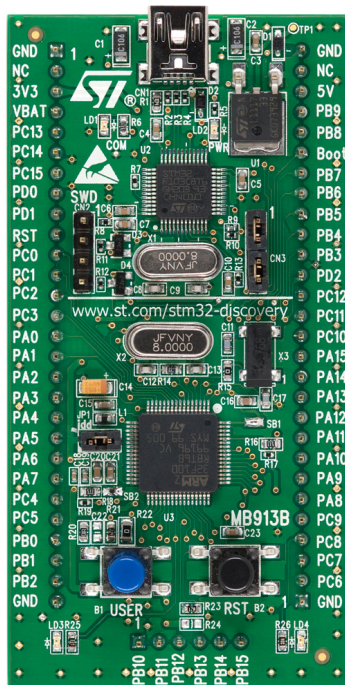


Obrázek 6.4: Schéma umístění prvků na platformě.

6.3.1 Řídící jednotka

Řídící jednotku pro prototyp tvoří vývojový kit STM32VLDISCOVERY, který byl zvolen převážně pro výhody vysoké připojitelnosti periférií různých typů. Vývojový kit STM32 obsahuje paměti 128kB flash 8kB RAM a napájen může být pomocí USB kabelu nebo externího zdroje o napětí 5 V nebo 3,3 V [26].

Na desce kitu je umístěno 52 konfigurovatelných výstupů, které mimo jiné umožňují i komunikaci s dalšími prvky na sběrnicích I²C, SPI a UART tedy s jedněmi z nejpoužívanějších typů u komunikace u senzorů obecně. Kompletní blokové schéma je vyobrazeno na obrázku 6.9.



Obrázek 6.5: Vývojový kit STM32VLDISCOVERY [26]

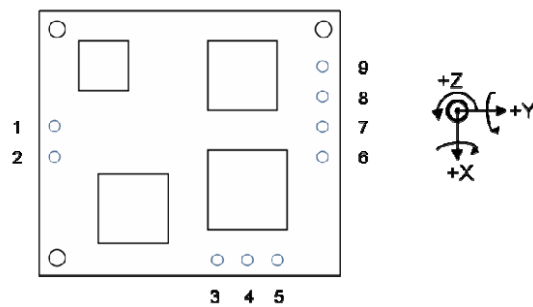
Vývojový kit také poskytuje řadu časovacích jednotek a šestnácti-kanalový převodník analogového signálu na digitální informace. Tento vývojový kit je dobře využitelný jako řídicí jednotka pro platformu quadrotoru, kde plní funkci řízení na základě dat ze senzorů a poskytuje dostatečný výkon pro základní výpočetní algoritmy. V případě potřeby využití výpočetně náročnějších algoritmů, například zpracovávání dat z LIDARu nebo zpracovávání obrazových dat, bude potřeba připojit výkonnější sekundární jednotku dedikovanou pro tyto výpočty nebo na tuto jednotku případně převést celé řízení platformy.

Vývojový kit STM32 je umístěn ve středu platformy, pro optimální možnost připojení ke všem periferiím a mimo proudy vzduchu od vrtulí.

6.3.2 IMU jednotka

Jednotka CHR-6D, měřící náklony a změny náklonů, je umístěna ve střední části platformy u řídicí jednotky. Jednotka je napájena napětím 3,3 V poskytovaným řídicí jednotkou a datově je přepojena přes UART rozhraní (na řídicí jednotce je využita sběrnice USART1). Připojení jednotky je následující:

- Pin 1 - Přívodní napětí 3,3 V
- Pin 2 - Uzemnění
- Piny 3 až 7 - piny pro nahrání nového firmwaru (na platformě nejsou zapojené)

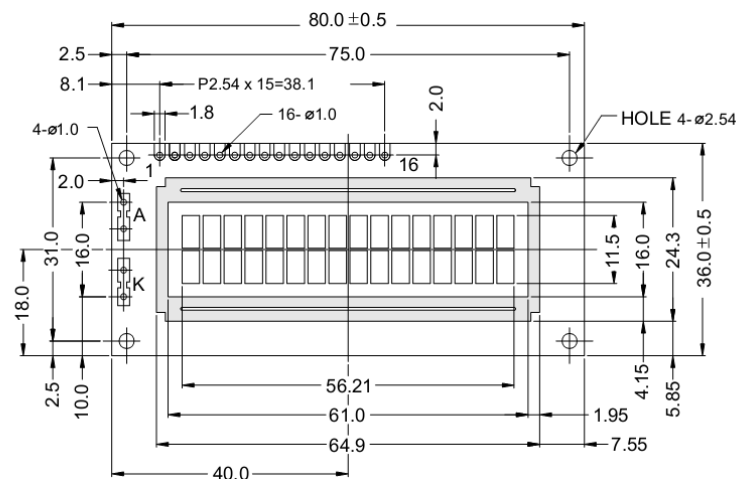


Obrázek 6.6: IMU jednotka CHR-6D [22]

- Pin 8 - UART-RX (pin pro příjem dat)
- Pin 9 - UART-TX (pin pro vysílání dat)

6.3.3 LCD panel

LCD panel pro zobrazování informací přímo na platformě je umístěn na okraji střední části platformy. Panel je napájen napětím 5 V a je řízen kanálem výběru registrů (RS) - připojen na pin PC6 řídicí jednotky, kanálem povolení operace (E) - připojen na pin PC7 a čtyřmi bitovými kanály připojenými na piny PB12 až PB15.



Obrázek 6.7: Schema LCD panelu [20]

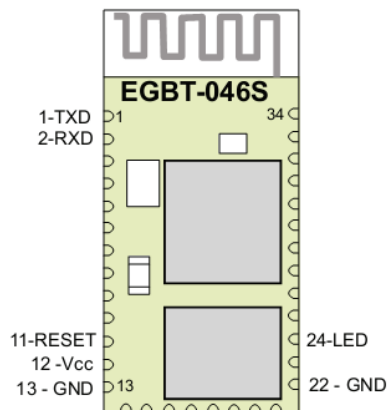
Popis připojení pinů na LCD panelu:

- Pin 1 - Uzemnění
- Pin 2 - Přívodní napětí 5 V
- Pin 3 - Úprava kontrastu (uzemněno)
- Pin 4 - Kanál pro výběr registru - RS
- Pin 5 - Výběr operace čtení nebo zápisu (uzemněno)
- Pin 6 - Kanál povolení operace - E
- Piny 7 až 10 - čtyři datové piny pro osmi-bitovou komunikaci (uzemněno)

- Piny 11 až 14 - čtyři datové piny pro čtyř-bitovou komunikaci
- Piny 15 a 16 - připojení napájení (+/-) podsvícení panelu

6.3.4 Bluetooth module

Bezdrátová komunikace platformy s okolím je zajišťována Bluetooth modulem typu EGBT-04. Tento Bluetooth modul umožňuje napájení napětím 3,3 V a datovou komunikaci na UART rozhraní. Modul je připojen k řídicí jednotce na druhou UART sběrnici (USART2) a je umístěn na okraji střední části platformy.



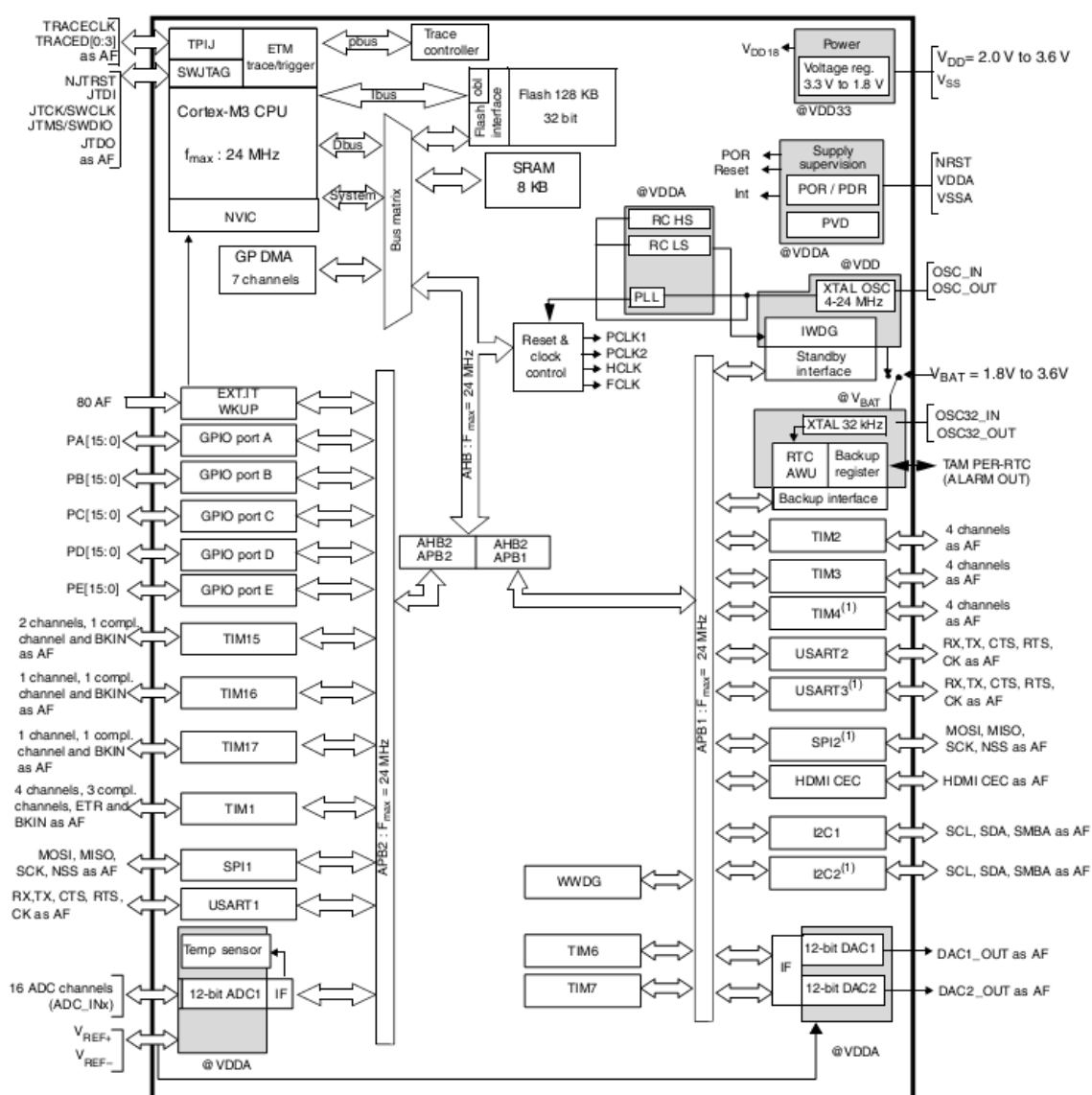
Obrázek 6.8: Schema Bluetooth modul [15]

Připojení vývodů modulu:

- Pin 1 - UART-TX
- Pin 2 - UART-RX
- Pin 11 - Reset
- Pin 12 - Přívodní napětí 3,3 V
- Pin 13 a 22 - Uzemnění
- Piny 25 až 33 - konektory s možností konfigurovatelného I/O (na platformě nejsou nezapojené)
- Pin 34 - Kanál CMD pro zasílání AT příkazů (nezapojeno)

6.3.5 Regulátory motorů

Regulátory motorů jsou upevněny na vzpěrách motorů, kvůli délce kabelů k motorům a také pro zabránění jejich přehřívání. Na platformě jsou využity regulátory dodávané společně s motory MIG RC 400/10 od firmy Megamotor. Regulátory umožňují konfigurování parametrů jako aktivní brzdění a způsoby časování, ale konfigurace probíhá změnou PWM střídy podle toho jak zrovna motor „pípá“, což se kus od kusu liší. Připojení regulátorů k řídicí jednotce je tří-pinovým konektorem, kde hnědá značí zem, červená je nezapojena a oranžová je připojena na PWM výstup.



Obrázek 6.9: Blokový diagram platformy STM32VLDISCOVERY [26]

6.4 Softwarová struktura

Software platformy je psán v jazyce C a kompilován GCC překladačem s podporou ARM architektury (konkrétně překladačem arm-atollic-eabi-gcc). Veškerý software pro platformu je psaný i překládáný pomocí vývojového nástroje Atollic TrueSTUDIO®/STM32 Lite v.2.0.1.

Funkce jednotlivých souborů je následující:

main(.c) Hlavní smyčka programu.

btcom(.c/.h) Manager bluetooth modulu.

pwm(.c/.h) PWM manager na ovládání motorů.

lcd(.c/.h) Manager LCD panelu.

chr6d(.c/.h) Manager IMU jednotky.

i2c(.c/.h) I²C komunikátor.

adc(.c/.h) Manager pro převodník analogického signálu na digitální.

6.4.1 **main(.c)**

Soubor **main.c** obsahuje hlavní řídicí smyčku, která kontroluje nová data ze senzorů a vypočítává podle nich výkon motorů. Součástí řídicí smyčky je i případné vypisování informací na LCD panel a odesílání požadovaných informací (jak pravidelných tak jednorázových) na bluetooth modul. Kromě hlavní smyčky obsahuje main.c následující funkce:

Err(char* txt) Výpis chyby na LCD panel a zablokování programu. Text chyby by neměl být delší než 16 znaků, protože zbytek bude ořezán. Po vypsání zprávy se software pokusí aktivovat BreakPoint() a upadne do věčného cyklu.

BreakPoint(void) Pomocná funkce pro ladění - TrueSTUDIO Lite v.2.0.1 umožňuje jen jeden breakpoint na celý kód, tak jsem vytvořil funkci, která ho obsahuje a je možné ji volat kdekoli, kde je potřeba.

Delay(uint32_t n) Aktivní čekání *n* milisekund. Funkce využívá interních systémových hodin a cyklů, dokud neuplyne stanovený čas.

motorsStartUp(char bool) Inicializace regulátorů motorů. Vstupní parametr určuje, zda se má provést zrychlená inicializace (která trvá 9 sekund) nebo standardní s testem motorů (s celkovým trváním 11,5 sekundy). Během nastavování regulátorů motory „pípají“, čímž informují o typu nastavení. Test motorů probíhá postupným spouštěním motorů a současnou indikací příslušnou diodou a informací na LCD panelu. Pokud je program v debug módu, je po testu pozastaven.

showBalance(int dioda,int power) Funkce pro nastavování diod, kde první parametr určuje číslo diody, pro kterou se má nastavení aplikovat a druhý parametr nastavuje 0 - nesvítí a interval (1-50) - svítí, přičemž 1 - svítí naplno a 50 - svítí tlumeně.

LedManager(void) Funkce na automatický management diod, je volána při systémovém tikku hodin a intenzitu diody emuluje pomocí pulzně šířkové modulace.

GPIOSetup(void) Nastavení portů pro diody - defaultně se používají porty: PC0, PC1, PC2 a PC3.

int fuzzyControl(float variable, int alfa, int beta, int high, int low, int zero) Funkce pro fuzzy výpočet dle obrázku 5.3. Funkce jako vstup přijímá hodnotu *variable*, pro kterou se vypočítává příslušný výsledek. Hodnoty *alfa* a *beta* pak značí hranice pravidel a hodnoty *high*, *low* a *zero* udávají výstupy jednotlivých pravidel. Výstupem funkce je celočíselná hodnota odpovídající výstupu po defuzzifikaci.

6.4.2 btcom(.c/.h)

Soubor **btcom(.c/.h)** obsahuje funkce pro komunikaci přes bluetooth modul. Defaultně se počítá s připojením bluetooth modulu k desce STM32 přes druhou UART sběrnici (USART2). Funkce souboru btcom.c:

bt_init() Funkce provádí inicializaci portů pro komunikaci a nastaví interní proměnné na výchozí hodnoty. Tato funkce by měla být zavolána jako první před dalším použitím tohoto souboru. Funkce by zároveň měla být volána před vstupem programu do hlavní smyčky.

bt_processPacket() Tato funkce slouží ke zpracování příchozích paketů. Paketu je nejdříve zkontrolováno, zda kontrolní součet v příchozím paketu odpovídá součtu vypočítaném z dat (tedy zda nedošlo k poškození paketu) a pokud odpovídá, je paket zpracován dle jeho typu. Názvy paketů jsou k hodnotám přiřazeny v souboru **btcom.h** dle tabulky 6.1.

BT_RX_DISPLAY	0xA1
BT_RX_TRANSMIT_DATA	0xA2
BT_RX_RECALIBRATE	0xA3
BT_RX_REQ_SETTINGS	0xA4
BT_RX_MOTORS	0xA5
BT_RX_MOTORS_BASE	0xA6
BT_RX_DIRECT_CONTROL	0xA7
BT_RX_POWER_SET_UP	0xA8
BT_RX_ROTATION_CORRECTION	0xA9

Tabulka 6.1: Pojmenování příchozích paketů.

uint8_t bt_clcChecksum(bt_pckType type, uint8_t len, uint8_t data[]) Funkce slouží pro výpočet kontrolního součtu. Jako vstup funkce přijímá typ paketu (*type*), délku dat v paketu (*len*) a samotná data, které paket obsahuje (*data*). Kontrolní součet je následně vypočítán jako součet všech vstupních parametrů a hlavičky paketu. Výsledná hodnota je poté ořezána na rozsah 0-255.

bt_send*() Funkce začínající jako „bt_send“ vyjma *bt_sendPacket* slouží k odesílání paketů definovaných v kapitole 7.3.2.2. Funkce nemají přímé vstupy a k samotnému odesílání používají právě funkci *bt_sendPacket*. Názvy paketů jsou k hodnotám opět přiřazeny v souboru **btcom.h** dle tabulky 6.2.

bt_sendPacket(bt_pckType type, uint8_t data[], uint8_t len) Tato funkce provádí přípravu paketu k odeslání, jako parametry přijímá typ paketu (*type*), data, která má paket obsahovat (*data*) a délku dat v paketu (*len*). Funkce nejdříve zkontroluje, zda se již odeslaly všechny předchozí pakety a případně počká. Pokud jsou předchozí pakety odeslány, nahraje funkce do odchozího bufferu data paketu, připojí kontrolní součet a nastaví příznak, že je možné data odeslat.

USART2_IRQHandler() Funkce pro obsluhu přerušení na komunikačním portu s bluetooth modulem. Tato funkce je aktivována v případě příchozích dat nebo požadavku

BT_TX_DATA_ANGLES	0x80
BT_TX_DATA_GYROS	0x81
BT_TX_DATA_ACCELS	0x82
BT_TX_SETTINGS	0x83
BT_TX_MOTOR_REPORT	0x84
BT_TX_MOTOR_BP_REPORT	0x85
BT_TX_POWER_COUNT_DATA	0x86
BT_TX_POWER_SET_UP	0x87

Tabulka 6.2: Pojmenování odchozích paketů.

na odeslání. Odesílání je prováděno postupným posíláním jednotlivých bytů a následným nastavením příznaku dokončení. Přijímání dat je prováděno pomocí stavového automatu, kde stavy 0-4 znamenají příjem hlavičky, typu paketu a délky dat a stav 5 znamená příjem samotných dat paketu a kontrolního součtu. Pokud je ve stavech 0-2 přijat jiný než očekávaný znak, je daný paket zahozen a stav nastaven zpět na 0. Po přijmutí plné délky paketu je paket odeslán funkcí *bt_processPacket()* na zpracování.

6.4.3 pwm(.c/.h)

Soubor **pwm(.c/.h)** slouží jako knihovna pro přístup k ovládání motorů. Motory na platformě jsou řízeny regulátory, které nastavují výkon motoru podle vstupních stříd. Jednotlivé regulátory motorů jsou tedy připojeny na generátory pulzně-šířkové modulace (PWM) na desce STM32. Pro práci s PWM generátory obsahuje soubor pwm.c následující funkce:

PWM_Init() Funkce, která inicializuje hodiny pro periférii *A* a *B* a nastaví výstupní porty PWM generátorů na výstupy PA6, respektive PA7 pro první, respektive druhý regulátor a PB0, respektive PB1 pro třetí, respektive čtvrtý regulátor.

PWM_Config(unsigned char channel) Tato funkce slouží k nakonfigurování jednotlivých kanálů časovače. Vstupní parametr *channel* určuje, který z kanálů se má nakonfigurovat.

PWM_Set(unsigned char channel, int pwm) Funkce pro změnu výstupní vlny. Vstupní parametry *channel* a *pwm* určují, který kanál se nastavuje a jaká je nová délka pulzu. Hodnota parametru *pwm* by se měla pohybovat v rozmezí 0-1000, jinak bude do tohoto rozmezí ořezána.

6.4.4 lcd(.c/.h)

Soubor **lcd(.c/.h)** definuje základní funkce pro ovládání a používání LCD panelu. Soubor je napsán jako univerzální knihovna pro LCD panely s řadičem Hitachi HD44780 a je tedy možné s ním ovládat i LCD panely jiných rozměrů či tvarů. Pro používání LCD panelů jsou definovány následující funkce:

GPIO_LCD_Inicializace() Funkce na inicializaci portu pro připojení LCD panelu. Zapojení panelu k portům je definováno v souboru **lcd.h**, kde je defaultně nastaveno:

Pojmenování	Port	Popis
LCD_E	PC7	Vstup hodinového signálu
LCD_RS	PC6	Výběr registru
LCD_D4	PB12	Dataport 4
LCD_D5	PB13	Dataport 5
LCD_D6	PB14	Dataport 6
LCD_D7	PB15	Dataport 7

Tabulka 6.3: Defaultní připojení LCD panelu.

LCD_Init() Inicializační funkce LCD panelu - tato funkce by měla nastavit LCD panel do čtyř-bitového módu nezávisle na tom, zda-li panel již byl do čtyř-bitového módu přepnut nebo zda je stále v módu osmi-bitovém.

PosliPrikazLCD(uint8_t cmd) Základní funkce pro aplikaci osmi-bitových příkazů. Příkaz je rozdělen na dvě části a odeslán pomocí funkce *Posli4bitLCD*

Posli4bitLCD(uint8_t cmd) Tato funkce přijímá osmi-bitové číslo a podle horních čtyř bitů nastaví výstupní porty pro LCD na příslušné hodnoty.

LCD_CursorHome() Pomocná funkce na nastavení kurzoru LCD panelu do výchozí pozice.

LCD_Clear() Pomocná funkce pro vymazání obsahu LCD panelu.

PosliRadekLCD() Pomocná funkce pro přepnutí se na další řádek.

LCDfullprint(char* txt1, char* txt2) Funkce pro výpis textu přes dva řádky LCD panelu, kde proměnná *txt1* obsahuje text prvního řádku a proměnná *txt2* obsahuje text řádku druhého. Funkce automaticky vyčistí obsah LCD panelu před tiskem a k tisknutí jednotlivých řádků využívá funkce *printLCD*.

printLCD(char* retezec) Tato funkce slouží pro tisk jednotlivých znaků ze vstupní proměnné *retezec* pomocí funkce *PosliZnakLCD*. Funkce tiskne znaky, dokud nenarazí na konec řetězce nebo nedosáhne limitu délky panelu definovaného v souboru *lcd.h*.

PosliZnakLCD(uint8_t znak) Funkce pro posílání jednotlivých znaků na LCD panel. Znak je rozložen do dvou příkazů a odeslán pomocí funkce *Posli4bitLCD*.

6.4.5 chr6d(.c/.h)

Soubor **chr6d(.c/.h)** funguje velmi podobně jako **btcom(.c/.h)** a má za úkol komunikaci s IMU jednotkou připojenou na prvním UART portu (USART1). Funkce pro inicializaci, příjem paketů, zpracování paketů, kontrolu paketů a odesílání paketů fungují analogicky k funkcím v kapitole 6.4.2. Konkrétní parametry přijímaných i odesílaných paketů jsou implementovány dle specifikace [22]. Soubor *chr6d.c* obsahuje navíc funkce:

chr6d_getUV() Funkce pro získání hodnot ze senzorů, kde *U* je nahrazeno znakem A pro data z akcelerometru a znakem G pro data z gyroskopu. Znak *V* je pak nahrazen

příslušnou osou, kterou chceme získat, tedy X, Y nebo Z. Data z jednotlivých os gyroskopu značí úhlovou rychlost ve stupních za sekundu v dané ose a data z jednotlivých os akcelerometrů jsou vyjádřeny jako metry za sekundu².

chr6d_getUVc() Funkce pro získání „čistých“ hodnot ze senzorů. Parametry *UV* se nahrazují stejně jako u funkce *chr6d_getUV*. Tato funkce, na rozdíl od předchozí, vrací nezpracované hodnoty, tedy šestnácti-bitové čísla odesílané IMU jednotkou.

int hasData() Tato funkce slouží k dotazu, zda přišly nové hodnoty, používá se v kombinaci s funkcí *dataRead()* a vrací hodnotu 0 pokud aktuální data již byla použita ke zpracování a hodnotu 1 pokud od IMU jednotky přišly nové data.

dataRead() Funkce pro označení dat příznakem přečtení (0). Příznak se do hodnoty 1 přepne automaticky s příchodem příslušného paketu od IMU jednotky.

6.4.6 i2c(.c/.h)

Soubor **i2c(.c/.h)** slouží ke zjednodušení komunikace na sběrnici I²C a obsahuje následující funkce:

I2CInit() Funkce sloužící pro inicializaci portů na komunikaci. Konkrétní porty, které se mají inicializovat jsou definovány v souboru *i2c.h*, kde defaultně se používá sběrnice I2C1 na portech PB6 a PB7.

I2CWrite(uint8_t DeviceAddr, uint8_t WriteAddr, uint8_t DataByte) Funkce pro odesílání dat, kde *DeviceAddr* specifikuje adresu příjemce, *WriteAddr* specifikuje adresu, kam uložit data a *DataByte* vyjadřuje vlastní data. Funkce využívá vnitřní knihovny STM32 a jejich funkcí *I2C_GenerateSTART*, *I2C_Send7bitAddress*, *I2C_SendData* a *I2C_GenerateSTOP*.

uint8_t I2CRead(uint8_t DeviceAddr, uint8_t ReadAddr) Funkce pro vyčítání dat, kde *DeviceAddr* specifikuje adresu dotazovaného a *ReadAddr* specifikuje umístění požadovaných dat. Funkce za využití příkazů *I2C_GenerateSTART*, *I2C_Send7bitAddress* a *I2C_SendData* odešle cílovému zařízení požadavek na data a následně je přečte pomocí *I2C_ReceiveData*. Celá komunikace je zase ukončena funkcí *I2C_GenerateSTOP*.

6.4.7 adc(.c/.h)

Soubor **adc(.c/.h)** slouží k nastavení a umožnění přístupu k převodníku analogového signálu na digitální informace. Soubor *adc.c* obsahuje funkci:

ADC_Inicializuj() Funkce, která inicializuje porty pro příjem analogového signálu. Defaultně tato funkce využívá prvního ADC převodníku (ADC1) a portů PC0, PC1, PC2 a PC3. Naměřené hodnoty se pomocí přímého přístupu do paměti (DMA) zapisují do pole *ADC_Senzory* s rozsahem šestnáct bitů.

Kapitola 7

Komunikace s okolím

Platforma quadrotoru je sice vyvíjena za účelem testování autonomních funkcí, ale důležitou součástí platformy tvoří i prvky na komunikaci s okolím. Komunikace s okolím představuje významný přínos nejen při ladění a testování, ale i při samotném provozu, kdy je vhodné, aby platforma mohla sdělovat svůj stav a případně i poskytovat dále měřená data ze senzorů. Aktuálně platforma využívá vizuální, akustické i datové komunikační prostředky.

7.1 Vizuální komunikace

Vizuální komunikace slouží především ke sdělování základních stavových informací o běhu programu.

7.1.1 Informační diody

Platforma aktuálně využívá osmi diod na rychlé poskytování základních informací. Jedná se o čtyři LED umístěné na vývojové desce STM32 pro poskytování informací, zda je deska v provozu, zda se využívá přístup do paměti a dvou diod, které aktuálně slouží k upozornění na chyby v přijímané komunikaci či chyby ve stavu programu.

Další čtyři diody jsou umístěné u jednotlivých motorů a slouží k poskytování informací o aktuálních změnách ve výkonu, což je vhodné především při ladění algoritmů.

7.1.2 Display

Platforma obsahuje 16x2 LCD Display modul, tedy dvouřádkový display o celkové zobrazovací kapacitě 32 znaků. Display obsahuje řadič Hitachi HD44780 a umožňuje zobrazování alfanumerických znaků a má možnost komunikovat v osmi či čtyř-bitovém módu.



Obrázek 7.1: Display umístěný na platformě.

Platforma má v rámci softwaru implementovanou knihovnu na ovládání displaye a umožňuje nejen komunikaci v rámci jednotlivých příkazů, ale i tisk celých řetězců a dá se tedy snadno využít k ladícím výpisům interních hodnot nebo přímo zpráv oznamujících například chybu a její důvod nebo oznámení úspěšně dosažených stavů.

7.2 Akustická komunikace

Akustická komunikace se většinou využívá především z bezpečnostního hlediska. Motory, využívané na platformě, během konfigurace (pokaždé před rozjezdem) už samy vydávají pískavé zvuky a během provozu celá platforma provozuje hluk v úrovni 90dB.

7.3 Datová komunikace

Datová komunikace platformy s okolím je nutná především ve fázích vývoje, kdy je potřeba mít přehled o hodnotách, se kterými quadrotor pracuje, ale je vhodná i ve fázi provozu, kde quadrotor může předávat údaje dalším prvkům k dodatečnému zpracování a vyhodnocení.

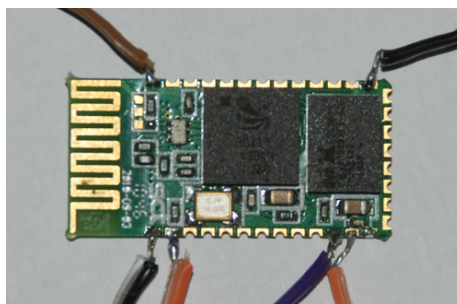
7.3.1 Drátová datová komunikace

Drátová datová komunikace je využita převážně v rámci platformy pro komunikaci s externími čidly a senzory, ale platforma umožňuje i připojení s okolím například přes UART sběrnici k paralelnímu sledování komunikace se senzory nebo přímo k odběru dat z quadrotoru.

Během fáze vývoje a ladění se také dá využít připojení přes USB kabel. Vývojový kit STM32 využitý na platformě obsahuje USB řadič a je tedy možné přímo sledovat činnost programu. Připojení přes USB kabel však má nevýhodu vzdálenosti a váhy USB kabelu a je tedy vhodné pouze na testování stojící nebo mírně levitující platformy.

7.3.2 Bezdrátová datová komunikace

Bezdrátová datová komunikace je pro provoz výrazně vhodnější než komunikace drátová, protože poskytuje volnost pohybu. Za tímto účelem je platforma quadrotoru vybavena Bluetooth modulem se schopností přenosu až 2.1 Mb za sekundu.



Obrázek 7.2: Komunikační bluetooth modul.

Bluetooth modul je připojen k platformě přes UART a je naprogramován, aby přeposílal komunikaci oběma směry. Pro testovací i provozní účely jsem vytvořil jednoduchý komunikační protokol. Každá zpráva začíná ascii hodnotami znaků „btx“, poté následuje identifikační číslo paketu t následované číslem l vyjadřujícím počet datových bitů d , které jsou následovány jedním kontrolním bytem ch , ve kterém je uložen součet všech bytů paketu ořezaný na hodnotu 0-255.

'b'	't'	'x'	t	l	$d_0 \dots d_l$	ch
-----	-----	-----	---	---	-----------------	----

Tabulka 7.1: Vzor paketů.

7.3.2.1 Pakety přijímané platformou

Nastavení LCD panelu - 0xA1

Tento paket slouží k nastavení frekvence výpisu informací na LCD panel, kde první byte l_e určuje, zda je výpis informací zapnutý a byte l_f udává hodnotu frekvence zobrazení ve 100 ms.

'b'	't'	'x'	0xA1	2	l_e	l_f	ch
-----	-----	-----	------	---	-------	-------	----

Nastavení požadovaných dat - 0xA2

Tento paket slouží k oznámení, která data má platforma odesílat zpět - konkrétně d_{pr} vyjadřuje požadavek na data Pitch a Roll vypočítané IMU jednotkou, d_g vyjadřuje požadavek na data z gyroskopu a d_a na data z akcelerometrů.

'b'	't'	'x'	0xA2	3	d_{pr}	d_g	d_a	ch
-----	-----	-----	------	---	----------	-------	-------	----

Příkaz na recalibraci senzorů - 0xA3

Tento paket slouží ke spuštění recalibrace senzorů na vodorovnou polohu. Pokud hodnota $R = 1$ spustí se recalibrace gyroskopu, která trvá zhruba 2 až 3 sekundy a platforma by měla po celou dobu být ve stabilní poloze. Hodnota $R = 2$ spustí recalibraci akcelerometrů, která by se měla provést v nejnižším cyklu programu.

'b'	't'	'x'	0xA3	1	R	ch
-----	-----	-----	------	---	-----	----

Požadavek informace o nastavení - 0xA4

Tento paket slouží k oznámení požadavku interního nastavení platformy a využívá se například při připojení klienta k platformě. Platforma na něj odpovídá paketem 0x83.

'b'	't'	'x'	0xA4	0	ch
-----	-----	-----	------	---	----

Povolení a startování motorů - 0xA5

Tento paket slouží k nastavení povolení a startování motorů. m_e určuje, zda je globálně na platformě povoleno používat motory a $m_s = 1$ spustí inicializaci motorů (pokud motory již inicializovány byly, je příkaz ignorován).

'b'	't'	'x'	0xA5	2	m_e	m_s	ch
-----	-----	-----	------	---	-------	-------	----

Nastavení základního tahu motorů - 0xA6

Tento paket slouží k nastavení základního tahu motorů - tedy společného tahu všech motorů. $b_e = 1$ znamená, že se využívá nastavení společného tahu motorů a byty $p_h p_l$ vyjadřují hodnotu 0-1000, která vyjadřuje úměrný výkon.

'b'	't'	'x'	0xA6	3	b_e	p_h	p_l	ch
-----	-----	-----	------	---	-------	-------	-------	----

Přímé ovládání motorů - 0xA7

Tento paket slouží k přímému vzdálenému ovládání, kde $c_e = 1$ vyjadřuje povolení ovládání a c_p a c_r určují rozdíly ve výkonu v osách Pitch a Roll.

'b'	't'	'x'	0xA7	3	c_e	c_p	c_r	ch
-----	-----	-----	------	---	-------	-------	-------	----

7.3.2.2 Pakety vysílané platformou

Přeposílání hodnot z IMU jednotky - 0x80,0x81,0x82

Tyto pakety slouží k přeposílání hodnot, které platforma získá z IMU jednotky 0x80 - znamená údaje o Pitch a Roll úhlech - každý uložen na dva byty, 0x81 označuje data ze tří os gyroskopu (po dvou bytech na osu) a 0x82 značí data z akcelerometrů - tři osy po dvou bytech na hodnotu.

'b'	't'	'x'	0x80	4	p_h	p_l	r_h	r_l	ch
-----	-----	-----	------	---	-------	-------	-------	-------	----

'b'	't'	'x'	0x81	6	z_h	z_l	y_h	y_l	x_h	x_l	ch
-----	-----	-----	------	---	-------	-------	-------	-------	-------	-------	----

'b'	't'	'x'	0x82	6	z_h	z_l	y_h	y_l	x_h	x_l	ch
-----	-----	-----	------	---	-------	-------	-------	-------	-------	-------	----

Informace o nastavení - 0x83

Tento paket slouží k informování klienta o vnitřním nastavení platformy a používá se hlavně při navázání kontaktu, aby klient s platformou měl nastavení sjednocené, jedná se o odpověď na paket 0xA4. V rámci bytu b je zakódování povolení (1) nebo zakázání (0) následujících nastavení: povolení zobrazování na LCD - b_0 , vysílání dat s úhly Pitch a Roll - b_1 , vysílání dat z gyroskopů - b_2 a vysílání dat z akcelerometrů - b_3 . Byte l_f pak udává nastavenou frekvenci obnovy informace na LCD panelu v jednotkách 100 ms.

'b'	't'	'x'	0x83	2	$b : 0000b_3b_2b_1b_0$	l_f	ch
-----	-----	-----	------	---	------------------------	-------	----

Informace o motorech - 0x84

Tento paket odesílá informace o nastavení motorů a to konkrétně m_e což vyjadřuje, zda je na platformě povoleno používání motorů a m_s , které vyjadřuje, zda již byly motory inicializovány. Tento paket se z platformy odesílá kdykoliv dojde ke změně některého z výše zmíněného.

'b'	't'	'x'	0x84	2	m_e	m_s	ch
-----	-----	-----	------	---	-------	-------	----

Informace o základním tahu motorů - 0x85

Tento paket informuje o základním tahu motorů, tedy zda je funkce povolena b_e a aktuální nastavení na motorech uložené na dva byty $p_h p_l$. Paket se odesílá vždy, když dojde ke změně tahu.

'b'	't'	'x'	0x84	3	b_e	p_h	p_l	ch
-----	-----	-----	------	---	-------	-------	-------	----

Kapitola 8

Vzdálené ovládání

Vzdálené ovládání slouží k usnadnění ladění a možnosti sledovat data z platformy. Aplikace pro vzdálené ovládání je napsána v Javě pro platformu Android a využívá bezdrátového přenosu přes Bluetooth a protokolu popsaného v kapitole 7.3.2.

8.1 Ovládání

Vzdálené ovládání umožňuje nastavení řady základních parametrů, zobrazování různých dat vysílaných platformou, nastavování motorů či algoritmů.

8.1.1 Sekce pozičních dat



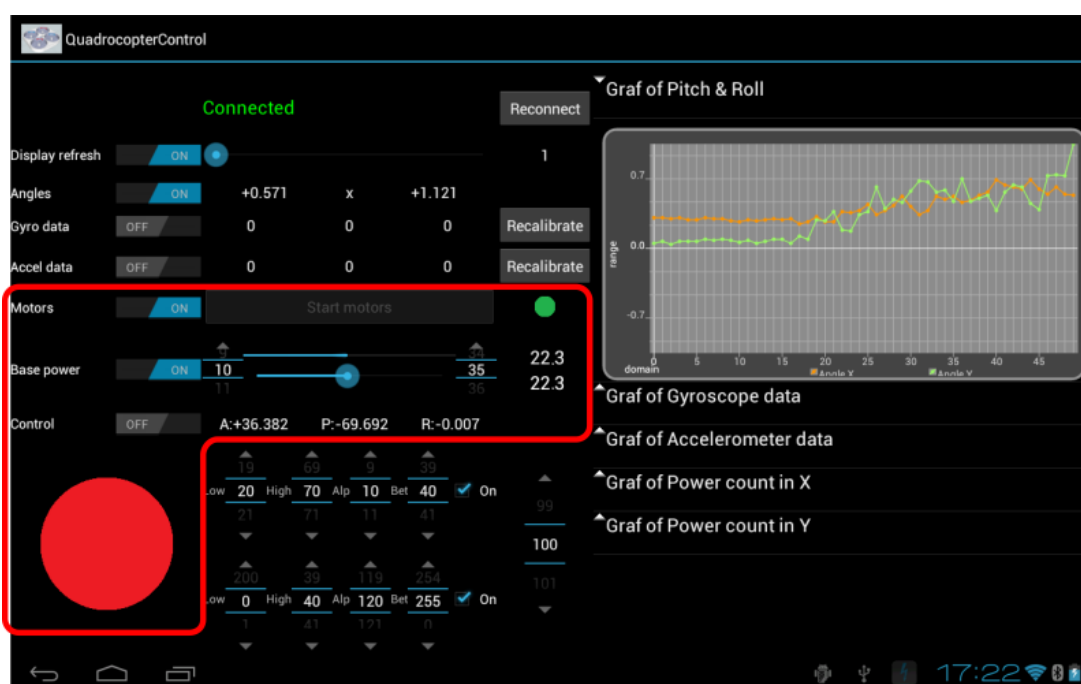
Obrázek 8.1: Vyznačená sekce pozičních dat a grafy náklonů a dat z gyroskopů.

Prvním prvkem v této sekci je nastavování obnovovací frekvence LCD panelu a případně jeho vypnutí. Další tři přepínače s názvy „Angles“, „Gyro data“ a „Accel data“

zapínají či vypínají přenos úhlů vypočítaných Kalmanovým filtrem nebo přímo surových dat z gyroskopu a akcelerometru IMU jednotky. Přenášená data se pak zobrazují v reálném čase v textové podobě u přepínačů, nebo je možné je zobrazovat na grafech ve vedlejším sloupečku.

Řádky „Gyro data“ a „Accel data“ také obsahují tlačítka „Recalibrate“, která umožňují kalibraci senzorů. Kalibrace gyroskopů probíhá v režii samotné IMU jednotky a může trvat až 3 vteřiny, během této doby se neaktualizují data a platforma nemusí odpovídat na vzdálené příkazy. Kalibrace akcelerometrů se provede okamžitě nastavením aktuálních hodnot jako hodnot určujících vodorovnou rovinu.

8.1.2 Ovládání motorů



Obrázek 8.2: Vyznačená sekce pro ovládání motorů.

Pomocí konzole je také možné ovládat motory, kde kvůli bezpečnosti je zaveden následující systém:

- Motors ON/OFF - nastavuje globální povolení motorů, kde pokud je vypnuto, motory se deaktivují a další nastavení je nepřístupné.
- Start motors - startér motorů - po povolení motorů, pokud ještě nebyly nastartovány, je možné motory nastartovat, což umožní další ovládání, které je jinak blokováno.
- Barevné kolečko vedle tlačítka na startování je indikátor stavu motorů, kde červená značí vypnuté motory, oranžová povolené motory, které ještě nebyly nastartovány a zelená značí povolené nastartované motory.
- Base power ON/OFF - určuje, zda se má aktivovat společný tah motorů.
- Limity a posuvník Base power - určuje společný tah motorů v rozmezí 0 až 100 (pro roztočení motorů je většinu potřeba hodnota přes 10 a při hodnotách nad 40 platforma

vzlétá od země). Rolovací výběrové prvky u okrajů posuvníku nastavují limity pro snazší nastavování určitých hodnot. Nastavená hodnota je zároveň zobrazována jako spodní z dvojice čísel napravo od posuvníku. Progress bar nad posuvníkem a horní číslo napravo pak indikují aktuální výkon, který platforma opravdu využívá.

- Velké barevné kolečko ve spodní části pak slouží jako hlavní spínač, kdy šedá barva znamená, že motory nelze spustit (například, že nejsou povolené nebo nastavené). Zelená barva indikuje připravenost a po kliknutí na něj se motory spustí a barva se změní na červenou. Dalším kliknutím se motory zase vypnou a barva se vrátí do zelené.

Vzdálené ovládání také nabízí kontrolu motorů na základě náklonových dat přímo z Android zařízení. Aktuální hodnoty náklonů zařízení se zobrazují vedle přepínače Control ON/OFF, který celé ovládání zapíná a vypíná a určují Az - azimut rotace Yaw, Pi - určuje rotaci Pitch, tedy změnu výkonu předního a zadního motoru a R - určuje rotaci Roll, tedy změnu výkonu bočních motorů. Všechny hodnoty jsou uvedeny v jednotkách $^{\circ}$.

8.1.3 Testovací parametry

Ve spodní části levého sloupce se pak ještě nachází nastavování parametrů fuzzy řízení, konkrétně hodnoty Low a High, které vyjadřují aplikovaný výkon a hodnoty Alpha a Beta, které v prvním případě pravidel vyjadřují stupně pro aktuální náklon a v druhém případě změnu náklonu v čase. Jednotlivá pravidla je možné pro účely ladění zapínat či vypínat pomocí checkboxu za výběrem hodnot.



Obrázek 8.3: Vyznačená sekce pro nastavení stabilizace a grafy zobrazující vliv pravidel na tah motorů.

Vliv výstupu fuzzy pravidel je pak možné sledovat na grafech v pravém sloupečku, kde je možné si pro jednotlivé osy zobrazit aktuální náklon (oranžově), změnu v tazích motorů v závislosti na náklonu - využití první sady pravidel (zelená) a výslednou změnu v tazích motorů při využití sad pravidel pro aktuální náklon a změnu náklonu v čase (modrá).

Poslední prvek ve vyznačené sekci na obrázku 8.3 slouží k úpravě rozdílů tahů motorů v ose X a v ose Y , tedy slouží k nastavení korekce nechtěné rotace. Základní hodnota je 100, kdy se neprovádí žádná změna a korekci je možné provádět v rozmezí hodnot 0 až 200, kde změna o každou jednotku představuje změnu v rozdílu výkonu os o 0,2%.

8.2 Softwarová struktura

Softwarová struktura konzole pro vzdálené ovládání je rozdělena do několika tříd:

QuadrocopterControlActivity je hlavní třída provádějící nastavení a aktualizaci GUI a zpracovávající bezdrátovou komunikaci. Zde probíhá načtení a nastavení ovládacích prvků v levém sloupci a snímání interního senzoru pro případné ruční ovládání quadrotoru. Jednotlivé funkce pro odesílání zpráv jsou pak označeny prefixem „tx_send“ a příchozí zprávy jsou zpracovávány funkcí „zpracujZpravu“.

CustomExpendableListAdapter je třída pro management rozbalovacích částí na pravé straně obrazovky. Tato třída zařizuje optimalizace, kdy se jednotlivé obsahy rolet mají vykreslovat a kdy je vykreslovat potřeba není pro snížení výpočetní zátěže.

SimpleDynamicSeries, DynamicXYDatasource jsou pomocné třídy pro obecné zobrazování grafů. Tyto třídy se instancují pro každý nový graf a uchovávají předchozí hodnoty, ze kterých se graf následně vykresluje.

ExpandablesManager je třída zastupující jednotlivé rozbalovací části. Třída obsahuje položku „type“, kde je možné nastavit typ rolety. Dále ve třídě je možné nastavit roletě její název a dle typu pak bude načten příslušný soubor, ze kterého se načte samotná vizualizace rolety.

Struktura a funkčnost jednotlivých tříd je navržena tak, aby bylo možné snadno přidávat další rozbalovací části nejen s grafy, ale i s libovolnými dalšími prvky.

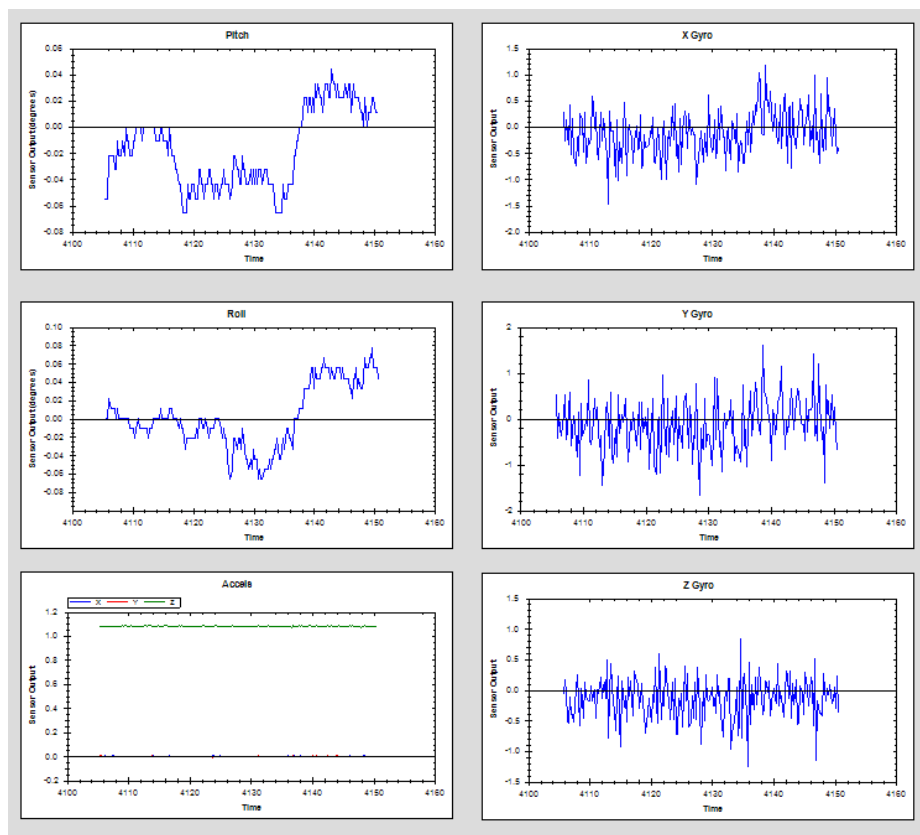
Kapitola 9

Konfigurace a testování

Software platformy je vyvíjen tak, aby mohl fungovat i při změnách hardwaru či konstrukce a většina funkcí tedy využívá konfigurační konstanty, pomocí kterých se program přizpůsobuje.

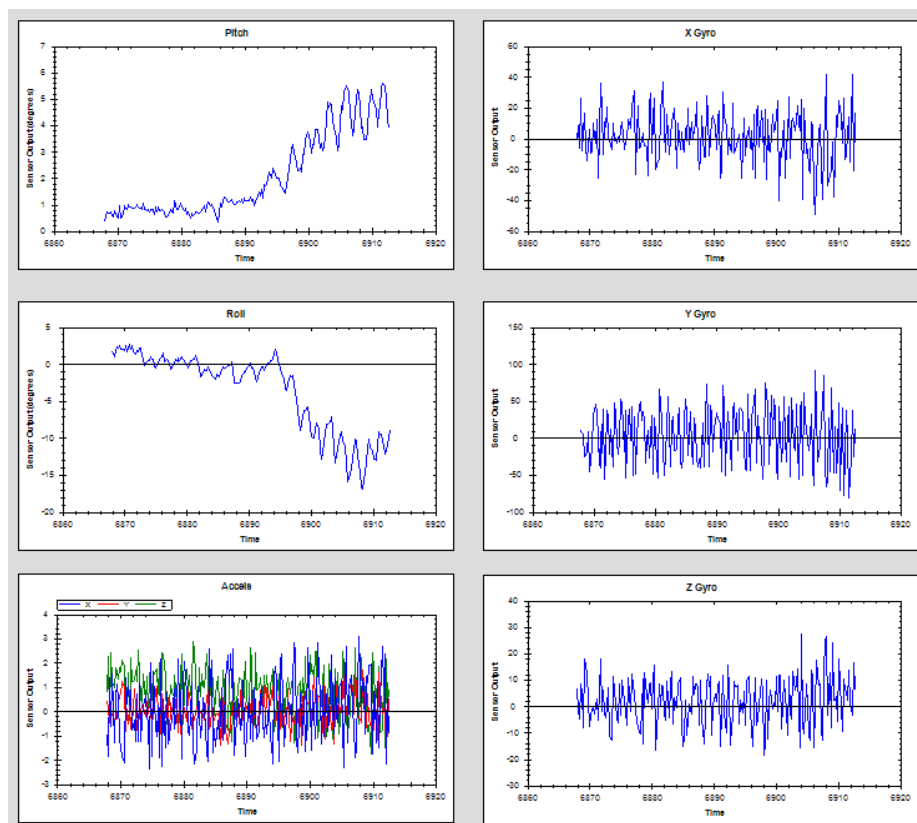
9.1 Zpracování informací o poloze

Pro testování na prototypu bylo využíváno dat získaných z inerciální měřicí jednotky, konkrétně z tříosého akcelerometru a gyroskopu, a jsou převáděny Kalmanovým filtrem na úhly Roll a Pitch.



Obrázek 9.1: Data senzorů z IMU jednotky při vypnutých motorech.

Platforma quadrotoru využívá k letu čtyři motory uchycené na kostře a tedy během letu „trpí“ nemalými vibracemi. K nekvalitě dat ze senzorů také přispívají změny napětí způsobované různými moduly a projevuje se i vliv vysílání bluetooth komunikace. Výstupní data bez běhu motorů jsou vyobrazeny na obrázku 9.1. Základní čištění dat je tedy nyní prováděno Parks-McClellan FIR filtrem s dolní propustí na 140 Hz a velikostí okna o 32 jednotkách. Kvůli vysokému vlivu vibrací na akcelerometry je v Kalmanově filtru kladen důraz především na data z gyroskopů a data z akcelerometrů se používají ke korekci v klidovém stavu. Vliv otřesů z motorů lze vidět na obrázku 9.2.



Obrázek 9.2: Data senzorů z IMU jednotky při spuštěných motorech.

Pro zmírnění otřesů je tedy na aktuální konstrukci samotná IMU jednotka umístěna na 5 mm vrstvě polyuretanové pěny a přívodní kabely jsou zajištěny proti přenosům otřesů. Při využití jiného typu motorů či jiné konstrukce, která nebude tolik přenášet otřesy, by bylo vhodné změnit velikost okna filtru, aby docházelo k menšímu zpoždění dat a quadrotor mohl na rozdílné náklony rychleji reagovat.

9.2 Nesouměrnosti modelu

Software platformy počítá i s různými nesouměrnostmi jako například rozdílný tah motorů nebo vychýlení těžiště. Korekce nesouměrností jsou definovány konstantami v souboru *main.c* softwaru platformy, případně jsou nastavitelné i za letu pomocí vzdáleného přístupu. Příkladem takových konstant jsou *X_POWER_CORRECTION* a *Y_POWER_CORRECTION*,

které nastavují rozdíl v tahu motorů v ose *X-Pitch* a v ose *Y-Roll* nebo nastavovací konstanta *ROTATION_CORRECTION*, čímž se mění rozdíl v tahu jednotlivých os a tím dochází ke změně celkových rotačních sil v ose *Yaw*.

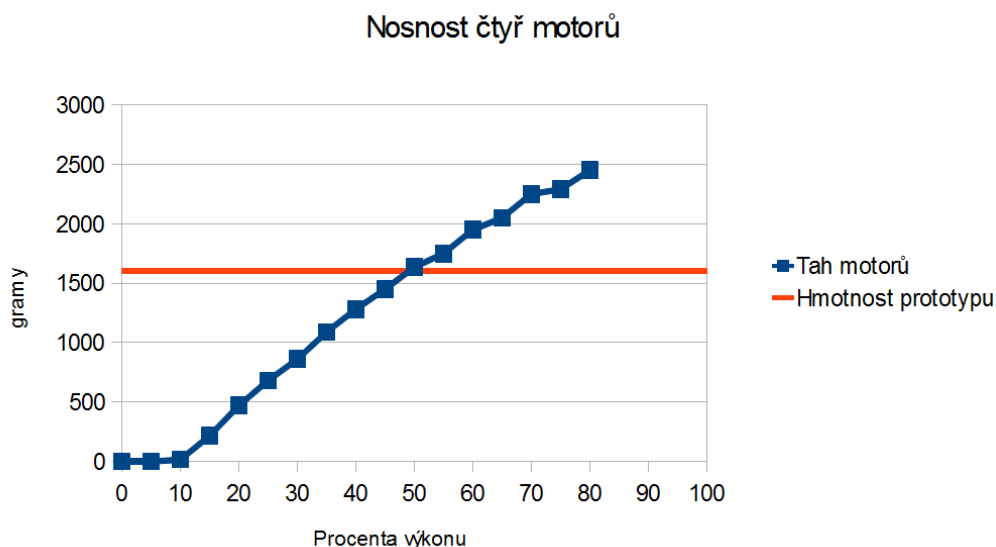
9.3 Testování

Testování probíhalo na sestrojeném prototypu o celkové váze 1647 gramů se čtyřmi motory MIG RC 400/10. Výrobce stanovuje doporučené váhy modelů pro tyto motory jsou 550 až 750 g. Součet doporučené váhy pro platformu je tedy 2200 g až 3000 g, avšak pro lepší manévrovatelnost a co nejvyšší reakční schopnosti je vhodné nechat výkonovou rezervu.

Výsledky testování jsou pouze orientační a pro přesné měření by bylo potřeba prototyp testovat v zařízení s úchytem o šesti stupních volnosti a v prostředí s konstantním tlakem vzduchu bez dalších proudů. Tedy v prostředí, kde se může prototyp volně pohybovat a zároveň může být přesně monitorován a zatěžován, a v prostředí s maximálním omezením proudů vzduchu nezávislých na prototypu i sekundárních proudů způsobovaných prototypem.

9.3.1 Test celkového tahu

Při aktuální konfiguraci platformy s třílistými vrtulemi GWS 8x4 motory vyvinou dostatečný vztlak pro let platformy nad zemí zhruba na polovičním výkonu. Experimentálně naměřené hodnoty jsou vyobrazené na obrázku 9.3.



Obrázek 9.3: Vyobrazení tahu motorů na platformě prototypu.

Nosnost byla testována zavěšením závaží o hmotnosti 2000 g na prototyp platformy. Závaží s platformou bylo postaveno na váhu a byla provedena řada testů se souměrným aplikováním různého výkonu motorů. Během testů byla zaznamenána hmotnost, se kterou závaží působilo na váhu. Maximální doporučené zatížení motorů je tedy 2500 g tedy zhruba

850 g nákladu pro aktuální prototyp. Optimální zatížení pro co nejvyšší manévrovatelnost je nosnost motorů pod 50% výkonu, tedy zhruba 1500 g, což je pod hmotností sestaveného prototypu.

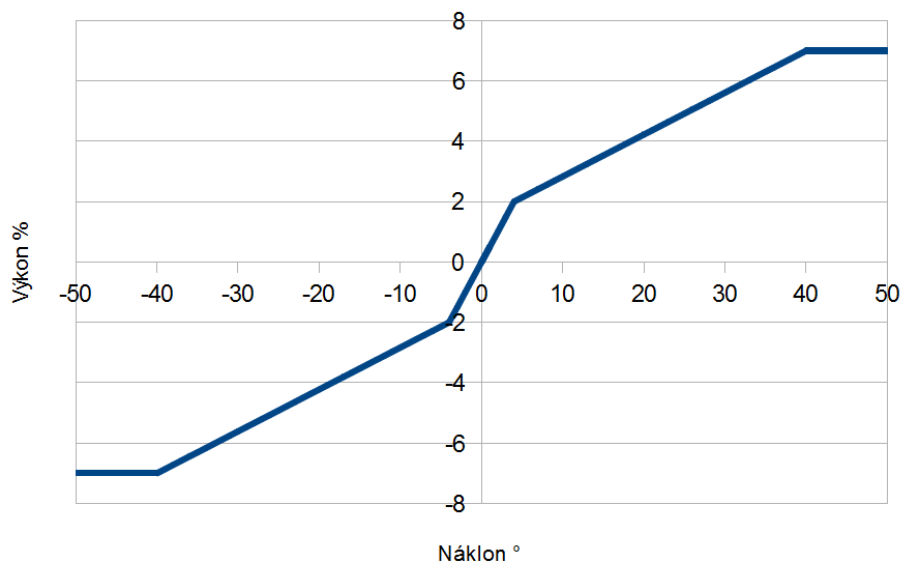
9.3.2 Tah motoru potřebný pro změnu náklonu

Tah motorů potřebný pro změnu náklonu je v ideálním prostředí libovolný nevyvážený tah. Tedy libovolné nesouměrnosti v tahu motorů vedou k vychýlení z aktuální pozice. Pokud však se quadrotor nachází v určitých vzdálenostech nad plochou, od které se proudy vzduchu odrážejí, dostává se do proudů, kde je k vychýlení směrem ku rovnováze potřeba rozdíl v tahu vyšší než určitá hodnota.

Dalším aspektem pro změnu náklonu je požadovaná rychlost změny. Rychlost změny náklonu je úměrná rozdílu v tahu motorů a pro platformu je samozřejmě jeden z požadavků co nejrychlejší stabilizace. Testy jsem tedy prováděl se zaměřením na nalezení optimálních parametrů pro co nejrychlejší změnu ku rovnovážné poloze za podmínky, že nebude docházet ke zbytečnému převažování do protilehlé polohy.

Výsledkem experimentů s modelem jsem dospěl k parametrům fuzzy pravidel (popsaných v kapitole 5.1.2), kdy výstupem pravidla *LOW* je změna výkonu o 2 % a pravidla *HIGH* o 7 % celkového výkonu. Zároveň úhly α a β byly stanoveny na 4° potažmo 40° náklonu v konkrétní ose. Celková závislost v rozsahu -50° až 50° je vyobrazena na grafu 9.4.

Závislost změny výkonu na náklonu



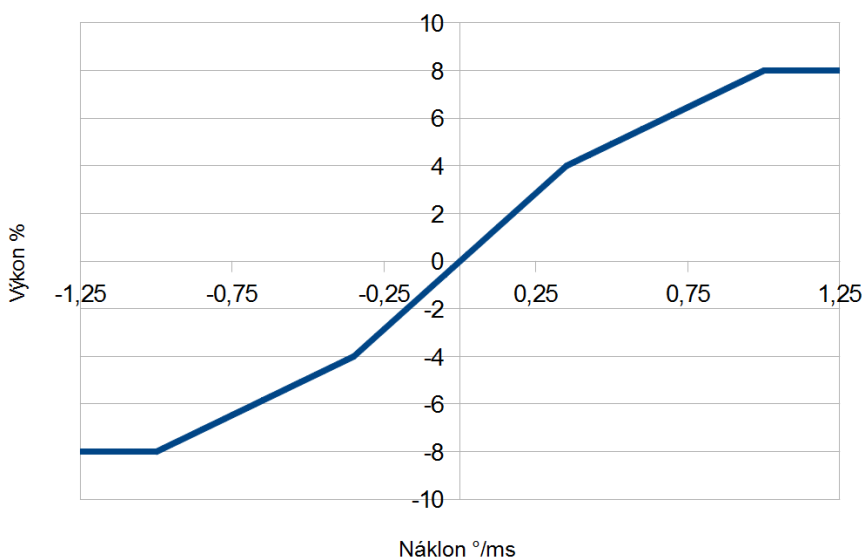
Obrázek 9.4: Graf závislosti rozdílu výkonu jednoho motoru na náklonu v příslušné ose.

Tyto experimentálně zjištěné hodnoty jsou samozřejmě závislé na „rozpětí“ quadrotoru, jeho hmotnosti a výšce těžiště. Pro použití na modelech jiných parametrů by mělo dojít k jejich úpravě případně nahrazení novými příslušnými hodnotami.

9.3.3 Tahy motorů pro stabilizaci

Pravidla z předchozí kapitoly samy o sobě pro stabilizaci nejsou vhodné, jelikož neuvažují hybnost platformy. K tomu účelu slouží druhá sada pravidel, která reaguje na aktuální rychlost změny náklonu. Experimentálně jsem došel na hodnoty *LOW* - 4 % a *HIGH* - 8 % celkového výkonu a α - $0,35^\circ/ms$ a β - $1,0^\circ/ms$.

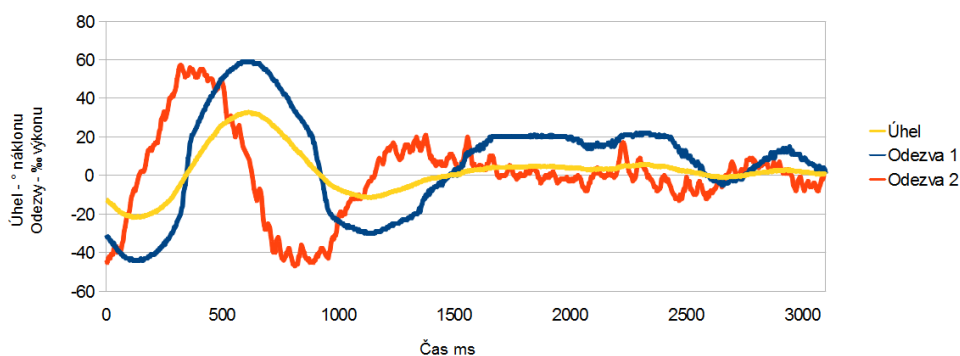
Závislost změny výkonu na změně náklonu



Obrázek 9.5: Graf závislosti rozdílu výkonu jednoho motoru na rychlosti změny náklonu.

Při těchto hodnotách dochází k tlumení rychlosti pohybu a platforma je schopná se stabilizovat. Odezvy fuzzy pravidel je možné vidět na příkladu náklonů na obrázku 9.6.

Odezvy fuzzy řízení



Obrázek 9.6: Graf odezvy fuzzy pravidel na náklon v jedné ose.

Kapitola 10

Závěr

Výstupem této práce je koncept a prototyp quadrotorové platformy pro účely výuky a testování, kde hlavním zaměřením je poskytnutí co nejvyšší modularity, aby bylo možné testovat různé prvky a konfigurace.

Koncept quadrotoru je směřován na obecný typ konstrukce, rozbor typů motorů a jejich napájení. Součástí konceptu je i rozbor možností využití různých typů senzorů pro získávání informací o poloze a vztahu platformy k okolí. Dále jsou v rámci konceptu rozebrány způsoby řízení stabilizace na základě PID a Fuzzy regulátorů.

Prvky z konceptu jsou pak aplikovány na sestrojený prototyp quadrotoru řízený jednotkou STM32VLDISCOVERY, jež využívá softwaru psaného v jazyce C a umožňuje připojení řady senzorů, například na sběrnících SPI, UART nebo I²C a případně i konverzi analogového na digitální signál. Prototyp dále umožňuje, za účelem ladění algoritmů, zobrazovat informace na dvouřádkovém LCD panelu a je schopen bezdrátové komunikace prostřednictvím sítě Bluetooth. Pro snazší ladění a sledování provozu quadrotoru byla také vyvinuta konzole pro vzdálený bezdrátový přístup v jazyce Java pro systém Android. Tato konzole umožňuje sledování informací ze senzorů v reálném čase a nastavování parametrů letu či přímo ovládání motorů. V rámci softwaru platformy je také implementována stabilizace pomocí fuzzy pravidel na základě dat o náklonu z inerciální měřící jednotky využívající tříosý akcelerometr a tříosý gyroskop.

Dle orientačních testů tahů motorů v kapitole 9.3.1 jsem stanovil limity na celkovou váhu na 2500 g pro maximální a 1500 g pro optimální zatížení. Potřebná změna výkonu pro stabilizaci prototypu je vyobrazená dvojicí grafů 9.4 a 9.5, zobrazující vztah změny aplikovaného tahu podle celkového a relativního náklonu.

Většina dnes dostupných nebo vyvíjených quadrotorů je úzce zaměřená na určitou činnost, nejčastěji sledování okolí a co nejdelší schopnost vznášet se nad cílem či co nejvyšší nosnost nákladu a nejdelší dráhu doletu. Quadrotor zpracovaný v této práci je oproti tomu zaměřen na univerzálnost a primární testování různých hypotéz před sestrojením specializovaného quadrotoru vhodného pro konkrétní činnost.

Pro řádné srovnání vlastního algoritmu stabilizace sestrojeného prototypu s algoritmy stabilizace jiných quadrotorů by bylo potřeba porovnání provádět na stejném modelu quadro-

toru. Obecně se ale dá předpokládat, že prototyp sestrojený pro tuto práci nebude výkonnější z důvodů typu konstrukce a zdaleka nedosáhne stabilizačních výkonů specializovaných quadrotorů. Zvýšení efektivity stabilizace prototypu by například bylo možné dosáhnout snížením roviny vrtulí blíže k těžišti a zvětšení rozpětí motorů. Pro sestrojení další verze platformy je pak také doporučeno použití lehčí konstrukce a provedení odstínění senzorů.

Literatura

- [1] *Zákon č. 49/1997 Sb. o civilním letectví*. Sbírka zákonů 225/2006 částka 74, (2006).
- [2] Battery energy. 2011, [Online; navštíveno 29.12.2011].
URL <http://www.allaboutbatteries.com/Battery-Energy.html>
- [3] Unique characteristics of Polymer Lithium Sulphur batteries. 2011, [Online; navštíveno 29.12.2011].
URL <http://www.oxisenergy.com/>
- [4] Apostolo, G.: *Illustrated Encyclopedia of Helicopter*. Bonanza Books, 1984.
- [5] Breuer, H.: Flying Inspector. In *Pictures of the Future*, Siemens, 2011, s. 64–66, [Online; navštíveno 23.12.2011].
URL http://www.siemens.com/innovation/pool/en/publikationen/publications_pof/pof_fall_2011/pof-2-2011-e-doppel.pdf
- [6] CTIE: Louis Charles Bréguet. 2002, [Online; navštíveno 31.12.2011].
URL <http://www.ctie.monash.edu.au/hargrave/breguet.html>
- [7] GmbH, B. S.: <http://www.bosch-sensortec.com/content/language1/downloads/BST-BMP085-DS000-05.pdf>. 2010, [Online; navštíveno 19.04.2012].
URL <http://www.bosch-sensortec.com/content/language1/downloads/BST-BMP085-DS000-05.pdf>
- [8] Guindehi, A.: Open Source Next Generation Multicopter. -, [Online; navštíveno 18.04.2012].
URL <http://ng.uavp.ch/moin/FrontPage>
- [9] Guindehi, A.: Wolferl Open Source QuadCopter. -, [Online; navštíveno 18.04.2012].
URL <http://uavp.ch/moin/FrontPage>
- [10] Huffman, J.; Chen, K.; Vong, B.: The Sharp GP2D12 Infrared Range Finder. 2008, [Online; navštíveno 25.12.2011].
URL <https://wiki.engr.illinois.edu/display/ae498mpa/The+Sharp+GP2D12+Infrared+Range+Finder>
- [11] INC., A. I.: Datron Scout Arial Reconnaissance System. -, [Online; navštíveno 12.04.2012].
URL <http://apogeeindustriesltd.com/scout.aspx>
- [12] Ji, X.; Lee, K.; Nazar, L.: A highly ordered nanostructured carbon–sulphur cathode for lithium–sulphur batteries. *Nature materials*, ročník 8, č. 6, 2009: s. 500–506.

- [13] Kalman, R.; aj.: A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, ročník 82, č. 1, 1960: s. 35–45, [Online; navštíveno 19.05.2012].
URL <http://www2.elo.utfsm.cl/~ipd481/Papers%20varios/kalman1960.pdf>
- [14] LaserMotive: UAV Power Links. 2010, [Online; navštíveno 28.12.2011].
URL <http://lasermotive.com/products/uav-power-links/>
- [15] e-Gizmo Mechatronix Central: Bluetooth Modules. [Online; navštíveno 24.04.2012].
URL <http://www.e-gizmo.com/KIT/images/EGBT-04/EGBT-045MS-046S%20Bluetooth%20Module%20Manual%20rev%201r0.pdf>
- [16] Mellinger, D.; Shomin, M.; Michael, N.; aj.: Cooperative grasping and transport using multiple quadrotors. In *Proceedings of the international symposium on distributed autonomous robotic systems*, 2010.
- [17] NASA: Aircraft Rotations. 2006, [Online; navštíveno 26.12.2011].
URL <http://www.grc.nasa.gov/WWW/K-12/airplane/rotations.html>
- [18] Nugent, T.: Laser-powered Quadrotor UAV Demo. 2010, [Online; navštíveno 28.12.2011].
URL <http://lasermotive.com/2010/11/12/video-of-laser-powered-quadcopter-endurance-flight/>
- [19] OmniVision: OV7670 VGA product brief. 2010, [Online; navštíveno 12.05.2012].
URL http://www.ovt.com/download_document.php?type=sensor&sensorid=74
- [20] Powertip: PC1602-C. [Online; navštíveno 22.04.2012].
URL http://www.powertip.com.tw/products_2.php?product_id=1171042363
- [21] Raza, S. A.; Gueaieb, W.: Intelligent Flight Control of an Autonomous Quadrotor. 2010, [Online; navštíveno 30.12.2011].
URL http://www.intechopen.com/source/pdfs/6587/InTech-Intelligent_flight_control_of_an_autonomous_quadrotor.pdf
- [22] Robotics, C.: CHR-6d Digital Inertial Measurement Unit - Product datasheet. 2010, [Online; navštíveno 19.04.2012].
URL http://www.chrobotics.com/docs/chr6d_datasheet.pdf
- [23] SHARP: GP2D120 General Purpouse Type Distance Measuring Sensor. -, [Online; navštíveno 25.12.2011].
URL http://sharp-world.com/products/device/lineup/data/pdf/datasheet/gp2d120_e.pdf
- [24] SHARP: GP2Y0A02K Long Distance Measuring Sensor. -, [Online; navštíveno 25.12.2011].
URL http://sharp-world.com/products/device/lineup/data/pdf/datasheet/gp2y0a02_e.pdf
- [25] s.r.o., J.: Střídavé motory MIG. [Online; navštíveno 12.05.2012].
URL www.megamotor.cz/

- [26] STMicroelectronics: STM32VLDISCOVERY - Discovery kit for the Value line STM32F100. 2010, [Online; navštíveno 20.04.2012].
URL <http://www.st.com/internet/evalboard/product/250863.jsp>
- [27] Technology, A. S. .: NAVSTAR GPS. [Online; navštíveno 11.03.2012].
URL http://www.spaceandtech.com/spacedata/constellations/navstar-gps_consum.shtml
- [28] Thrun, S.; Burgard, W.; Fox, D.: *Probabilistic Robotics*. MIT Press, 2005.
- [29] Wikipedia: PID controller. 2004, [Online; navštíveno 21.04.2012].
URL http://en.wikipedia.org/w/index.php?title=PID_controller&oldid=488159451
- [30] Wikipedia: Kalman filter. 2011, [Online; navštíveno 19.05.2012].
URL http://en.wikipedia.org/w/index.php?title=Kalman_filter&oldid=493211224
- [31] Wiora, G.: Principle of a sonar or radar distance measurement. 2005, [Online; navštíveno 26.12.2011].
URL http://en.wikipedia.org/wiki/File:Sonar_Principle_EN.svg
- [32] Zhang, X.; Fronz, S.; Navab, N.: Visual marker detection and decoding in AR systems: A comparative study. In *Proceedings of the 1st International Symposium on Mixed and Augmented Reality*, IEEE Computer Society, 2002, str. 97, [Online; navštíveno 10.05.2012].
URL <http://fieldrobotics.org/navgroup/papers/zhangFiducialComp.pdf>

Příloha A

Obsah CD

- ./quadrotor/Debug/vrtulnik.elf - přeložený firmware pro prototyp quadrotoru
- ./quadrotor/src/ - složka s kompletními zdrojovými kódy prototypu
- ./konzole/bin/QuadrocopterControl.apk - Aplikace konzole pro platformu Android
- ./konzole/src/ - složka s kompletními zdrojovými kódy konzole
- ./prototyp-dokumentace/ - složka s dokumentací k prototypu
- ./text_dp/ - složka se zdrojovým textem této práce